

An Approach to Strategic Capacity Planning Using Information Visualization

by

Robert J. Kelly

B.S. Electrical Engineering, Bucknell University, 1992

Submitted to the Sloan School of Management and the
Department of Electrical Engineering and Computer Science
In Partial Fulfillment of the Requirements for the Degrees of

Master of Business Administration
and
Master of Science in Electrical Engineering


In conjunction with the Leaders for Manufacturing Program at the
Massachusetts Institute of Technology
June 2002

© Massachusetts Institute of Technology, 2002. All rights reserved.


Signature of Author _____


MIT Sloan School of Management
Department of Electrical Engineering and Computer Science
May 10, 2002

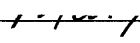
Certified by _____


Jim Hines
Senior Lecturer, Sloan School of Management
Thesis Advisor


Certified by _____

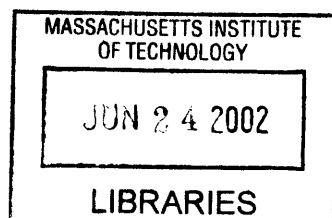

Duane S. Boning
Associate Professor, Electrical Engineering and Computer Science
Thesis Advisor

Accepted by _____


Margaret C. Andrews
Director of Master's Program, Sloan School of Management

Accepted by _____


Arthur C. Smith
Chair, Committee on Graduate Students
Electrical Engineering and Computer Science



BARKER

An Approach to Strategic Capacity Planning Using Information Visualization

by
Robert J. Kelly

Submitted to the Sloan School of Management and the Department of Electrical Engineering and Computer Science on May 10, 2002 in Partial Fulfillment of the Requirements for the Degrees of Master of Business Administration and Master of Science in Electrical Engineering

Abstract

The strategic capacity planning function must be a core competency for firms in capital-intensive, long equipment lead-time industries such as the semiconductor industry. Major investment decisions are made years in advance of forecasted demand, and involve a complex array of product design and manufacturing parameters. The high stakes and breadth of uncertainty require a well-orchestrated, comprehensive process of data analysis and management judgment that is the strategic capacity planning process.

The exploration and presentation of demand and capacity data are integral to the capacity planning process. The exploration activities are executed in a sequential, task-oriented manner involving manual data transcription and organization. The presentation of data is typically limited to standard graphing techniques such as line and bar graphs. Hence, the opportunity exists for alternative visualization tools to improve the information work of foraging for data, thinking about data, and acting on data.

This research effort involves the use of two distinct information visualization tools to augment the strategic capacity planning process. The What-If Visualization Tool is an Excel-based capacity calculator that facilitates the real-time, interactive manipulation of factors that affect capacity. This tool simplifies the current scenario exploration process and encourages active involvement in discovery. The Treemap display tool is a Java-based display methodology that vividly and concisely conveys highly detailed, hierarchical information in a two-dimensional map. This tool enhances the decision-making forums through its insightful, novel approach to data display.

We first examine the current literature supporting the information visualization field, and discuss the advantages to be derived from the proposed software tools given these constructs. Next is a description of the two visualization tools, including their designs, functionalities, and user interfaces. Finally, we illustrate the relevant advantages and shortcomings of these tools in the strategic capacity planning environment.

Thesis Advisor: Jim Hines
Title: Senior Lecturer, Sloan School of Management

Thesis Advisor: Duane Boning
Title: Associate Professor, Electrical Engineering and Computer Science

ACKNOWLEDGEMENTS

No words can express my gratitude to my wife, Theresa, and our twin girls, Kendall and Reiley. They have been my staunchest supporters, unfailingly loving even in the most difficult of circumstances. They are dearest to my heart.

I would like to thank Intel Corporation and the Leaders for Manufacturing (LFM) Program for granting me the privilege to develop intellectually, socially, and professionally in such a unique program at the Massachusetts Institute of Technology. Many thanks also to my LFM and Sloan peers, who challenged me to work diligently, solicit divergent viewpoints, and test uncharted waters.

I am greatly indebted to Dave Fanger, who encouraged me to think differently and welcome new insights. Dave's responsiveness and interest in my thesis work made the effort an enjoyable experience, and has altered the lens through which I see the world for the better. As Proust noted:

"The real voyage of discovery consists not in seeking new landscapes but in having new eyes."

Many thanks to the Strategic Capacity Planning group for the terrific internship experience, and the privilege of working with such talented and accommodating individuals. Thanks to Dan McKeon and Brian Kelly, for their encouragement over the course of the internship. Special thanks to Jeff Zino, for his patient tutelage and personal friendship during the internship experience. I could not have completed the project without his assistance. Thanks also to Nick Rose, who met every request for data, and was a good friend throughout.

I am grateful for the understanding and patience of my thesis advisors, Duane Boning and Jim Hines. Their help and encouragement provided me the fuel I needed in the final stretch. Special thanks to Duane for offering assistance and advice when I needed it the most.

Special thanks to Ben Shneiderman. Ben's generosity, patience, and genuine desire for advancing the field of information visualization have been a welcome breath of fresh air for my thesis work. Many thanks also to Bob Eberlein for his quick response to technical questions.

Finally, I would like to thank my parents, Robert and MaryAnn Kelly, for their love and support.

TABLE OF CONTENTS

1	INTRODUCTION	11
1.1	BACKGROUND	11
1.1.1	<i>Industry Background and Challenges</i>	<i>11</i>
1.1.2	<i>Strategic Capacity Planning Overview</i>	<i>12</i>
1.1.3	<i>Strategic Capacity Planning at Intel Corporation</i>	<i>14</i>
1.2	MOTIVATION FOR THIS WORK	16
1.3	APPROACH AND METHODOLOGY	17
1.3.1	<i>What-If Visualization Tool</i>	<i>17</i>
1.3.2	<i>Presenting Detailed, Hierarchical Data with Treemaps</i>	<i>18</i>
1.4	THESIS STRUCTURE	20
2	INFORMATION VISUALIZATION AS A FIELD OF STUDY.....	21
2.1	ORIGINS OF INFORMATION VISUALIZATION	21
2.2	VISUALIZATION AS A MEANS TO IMPROVE COGNITION	22
2.3	TRANSFORMING DATA INTO VISUAL INFORMATION	26
2.4	USABILITY CONSIDERATIONS	29
2.5	TREEMAPS: A DATA VISUALIZATION METHODOLOGY	32
2.5.1	<i>Treemap Overview.....</i>	<i>32</i>
2.5.2	<i>Display of High-Density Data.....</i>	<i>35</i>
2.5.3	<i>Squarification Algorithm.....</i>	<i>36</i>
3	THE WHAT-IF VISUALIZATION TOOL.....	39
3.1	FRAMEWORK FOR THE WHAT-IF VISUALIZATION TOOL	39
3.1.1	<i>Limitations of the Current What-If Process</i>	<i>40</i>
3.1.2	<i>Concept for the What-If Visualization Tool (WVT).....</i>	<i>43</i>
3.2	WSPW CALCULATION MODEL	45
3.2.1	<i>Semiconductor Manufacturing Supply Chain Overview</i>	<i>45</i>
3.2.2	<i>Incorporating Inventory Strategies into the Model</i>	<i>49</i>
3.3	WVT EARLY DESIGN DECISIONS: INTERACTIVE VENSIM	51
3.3.1	<i>Interactive WSPW Model Using Vensim 5.0</i>	<i>52</i>
3.3.2	<i>Data Proliferation and Effect on Model Interactivity</i>	<i>54</i>
3.3.3	<i>Interactive Model Insights and Subsequent Approach.....</i>	<i>57</i>
3.4	WVT ARCHITECTURE: EXCEL-BASED MODEL	57
3.4.1	<i>Illustration of WVT Interoperability.....</i>	<i>59</i>
3.5	WVT STRUCTURE AND FUNCTIONALITY	62
3.5.1	<i>WVT User Interface</i>	<i>62</i>
3.5.2	<i>Spreadsheet Calculations</i>	<i>65</i>
3.5.3	<i>Dynamic Scenario Exploration</i>	<i>67</i>
3.6	CHAPTER SUMMARY.....	68
4	DATA VISUALIZATION WITH TREEMAP 3.0.....	69
4.1	HUMAN INTERACTION MODEL – TREEMAP 3.0.....	69
4.2	THE TREEMAP DATA MACRO	72
4.3	TREEMAP DISPLAY OF LRP DATA.....	72

5	RESULTS AND ANALYSIS	75
5.1	WHAT-IF VISUALIZATION TOOL	75
5.1.1	<i>WVT Data Validation</i>	<i>75</i>
5.1.2	<i>WVT Usability Characteristics.....</i>	<i>77</i>
5.1.3	<i>Relationship Detection with the WVT.....</i>	<i>79</i>
5.2	TREEMAP DISPLAYS	80
5.2.1	<i>Emergence of a Dominant Design.....</i>	<i>81</i>
5.2.2	<i>Automating Treemap Generation.....</i>	<i>82</i>
5.3	SUMMARY	83
6	CONCLUSIONS AND RECOMMENDATIONS	85
6.1	SUMMARY OF FINDINGS	85
6.2	SUGGESTIONS FOR FURTHER RESEARCH.....	86

LIST OF FIGURES

Figure 1: LRP Process	15
Figure 2: Semiconductor Capital Spending.....	16
Figure 3: Treemap Display - Product Data Sized by WSPW and Organized by Fab	19
Figure 4: The Basic Boxplot.....	25
Figure 5: Human Interaction Model	27
Figure 6: Interactive Mapping	29
Figure 7: S-R Compatibility Example – Stovetop Configurations.....	30
Figure 8: Comparison of List and Tree Structures in Hierarchical Data Display	32
Figure 9: Treemap Representation	33
Figure 10: Treemap Display versus Pie Chart - 156 Data Elements Depicted	36
Figure 11: Squarification Algorithm Sequence.....	38
Figure 12: What-If Scenario Categorical Analysis	41
Figure 13: Current What-If Process - Human Interaction Framework.....	42
Figure 14: What-If Visualization Tool - Human Interaction Framework	44
Figure 15: Semiconductor Manufacturing - Throughput time and Yield Effects	46
Figure 16: Die Yield as a Function of Process Maturity	48
Figure 17: Semiconductor Supply Chain – Inventory Points.....	50
Figure 18: WSPW Model in Vensim 5.0 – Control Panel	52
Figure 19: WSPW Model in Vensim 5.0 – Model Structure	53
Figure 20: WVT Architecture	58
Figure 21: Sample VBA Code for WVT Interoperability Example.....	61
Figure 22: WVT User Interface.....	63

Figure 23: WVT User Interface - Cycle Quarter Selection Dialog	64
Figure 24: WVT User Interface - Field Population.....	65
Figure 25: Sample Spreadsheet Calculation Segment.....	66
Figure 26: Sample VBA Code for Parameters and Variable Population	66
Figure 27: WVT Dynamic Exploration.....	67
Figure 28: Treemap 3.0 User Interface.....	69
Figure 29: Treemap 3.0 - Dynamic Query Capability.....	70
Figure 30: Demonstration of Zoom Feature of Treemap 3.0 (base display Fig. 28).....	71
Figure 31: Treemap Display - Size by WSPW.....	73
Figure 32: Treemap Display - Size by Unit Demand	74
Figure 33: Demonstration of S-R Compatibility - WVT Example	78
Figure 34: WVT - Nonlinear Relationship Depiction	79
Figure 35: Treemap Paired Comparison.....	80

LIST OF TABLES

Table 1: Sample What-If Scenario Descriptions	40
Table 2: Illustration of Subscripted Arrays - Die Size by Product/Process	55
Table 3: Expanding Die Size Vector Length Using Offset Variable	56
Table 4: Sample Database Table: Product_Die_Size	59
Table 5: Query Output for WVT Interoperability Example	60
Table 6: Paired Comparison of WVT/LRP WSPW Calculations	76
Table 7: Treemaps in Use - Parameter Mix.....	81

This Page Intentionally Left Blank

1 INTRODUCTION

Strategic capacity planning is a critical competency of capital-intensive firms, particularly those in high-growth, high-clockspeed¹ industries where capital is needed for both the expansion and the refurbishment of the installed base. The semiconductor² industry is one such industry, and the background and challenges in this industry are introduced. The strategic capacity problem is then considered, with specific emphasis on the role the Strategic Capacity Planning (SCP) group fulfills within Intel. Most importantly, the motivation and goals for this work – to improve the visualization of complex strategic capacity data and scenarios – is offered. Finally, the organization of this thesis is presented.

1.1 BACKGROUND

1.1.1 Industry Background and Challenges

Growth in the semiconductor industry has been driven largely by decreases in device³ cost, increases in wafer⁴ processing capacity, and innovation in design and process technologies. In addition to demand growth, the transition to next-generation process technologies historically has required a 70/30 ratio of new to old equipment [10]. Hence, these advances have precipitated progressively higher demand for fabrication facilities and sophisticated manufacturing tools – tools that typically require significant capital expenditures and long procurement lead times. Because capacity planning is at the top of the plant planning hierarchy, choices about how much and what type of capacity to install have a strong influence on the firm's bottom line [12]. Excess capacity results in a higher allocated fixed cost, while insufficient capacity can lead to lost revenue and relinquished market share. As a result, successful companies in this industry are often differentiated by the effectiveness of their strategic capacity planning processes.

¹ Clockspeed: a term used to describe the rate of change undertaken by a firm (e.g. process innovation) [9].

² Semiconductor: a solid crystalline material with electrical conductivity between metals and insulators.

³ Device: a semiconductor product, such as a microprocessor, manufactured on wafers.

⁴ Wafer: a thin, disc-shaped slice of semiconducting material (e.g. silicon) from which chips are made.

Since the early 1970's the industry has been marching to the beat of Moore's Law¹, the seminal insight that the number of transistors² on a square area of silicon would double every 18 months, and that device cost would decrease proportionally. This has been accomplished with incredible advances in technology, particularly in the lithography tools that transfer the circuit pattern to the underlying silicon wafer. These tools have broken critical dimension barriers time and again, yielding a minimum feature size today on the order of tens of nanometers (one-billionth of a meter). Such enabling technologies command a steep price tag, often in the tens of millions of dollars apiece, and require firm purchase commitments two years in advance. Indeed, the total cost of a wafer fabrication facility has risen to over \$2 billion.

The industry has recognized that continued economic health depends upon better management of capital investment, including capital planning, procurement, and utilization. One strategy pursued to cope with the high fixed costs is to seek economies of scale through partnerships with other manufacturers. A manifestation of this focus has been the recent rise to prominence of semiconductor foundries, or contract manufacturers. While foundries now account for 16% of global chip production, Dataquest predicts that they will process 40% of total output by 2010 [8]. These firms ideally maximize equipment utilization by pooling demand requirements from their various customers. While this strategy may be optimal for some small-volume producers, firms such as Intel Corporation for which manufacturing is a core competency must rely upon the long-range planning process. Indeed, just as Fine [9] posits that supply chain design is the "meta" competency of the firm, creating the support framework for other competencies, long range planning, and specifically strategic capacity planning, may arguably be the "meta-meta" competency of the semiconductor manufacturing firm.

1.1.2 Strategic Capacity Planning Overview

The structure of a mature manufacturing enterprise at any given point in time reflects the accumulation of a series of prior resource allocation decisions. Decisions regarding which markets to pursue and which to abandon, which products to emphasize and which to phase

¹ Gordon Moore, co-founder of Intel, made his memorable observation in a speech in 1965.

² Transistor: an electrical device analogous to a "switch" – the building blocks of complex circuitry.

out, and which technologies to develop all work to shape the firm's physical assets, human capital, technological capabilities, and overall competitiveness [11]. The goal of strategic capacity planning then is to determine aggregate production quantities and the level of resources required to achieve these production goals. As Charles F. Kettinger noted: "My concern is with the future since I plan to spend the rest of my life there" [17]. Aggregate forecasts are important for two reasons. First, the aggregate forecasts are used in the strategic analysis of fixed commitments and requirements for new manufacturing and distribution facilities, new supply contracts or vendor agreements, and commitments to process and product technologies. Second, detailed forecasts can be derived in a sequential manner from the aggregate forecast by breaking it down based on market segmentation and manufacturing allocation strategies [22].

Macro planning begins with the forecast of demand. Demand forecasting is complex and multifaceted, not a simple matter of single prediction. Forecasts cover multiple time periods, multiple products, and in the semiconductor industry, often multiple manufacturing processes. As the aggregate planning methodology typically requires the assumption that demand is deterministic, demand forecasts are generally wrong. There is certain to be a random component of the demand that cannot be predicted exactly in advance. These uncertainties need to be taken under consideration, and indeed are addressed through both quantitative and qualitative forecasting methodologies. As mentioned above, competitive leverage can hinge on the firm's aptitude for maximizing information for, and minimizing the time required in, forecasting processes [22].

Quantitative forecasting methods are based on the assumption that the future can be approximated using mathematical forecasting techniques such as causal modeling or time-series modeling [12]. A causal model will attempt to identify what independent variables affect the dependent variable. For instance, a causal demand model may consider the effect of market share, market penetration, and GDP on forecasted demand. A time-series model, on the other hand, uses past results to predict the future. For instance, a best-fit regression line may be derived using historical demand for a certain product. The line could then be extrapolated to estimate future demand.

Qualitative forecasting methods attempt to develop likely future scenarios by using the expertise of people, rather than predictive mathematical models [12]. A scenario evaluation process is typically employed to provide a context for thinking clearly about the impossibly complex array of factors that affect any long range planning decision. It gives decision makers a common language for discussing these factors, and usually takes the form of a “what if” story, each identified uniquely [23]. For instance, a semiconductor firm may ask: “What will be the impact on our demand if the corporate IT personal computer upgrade period extends from 4 to 5 years? How will our capacity be impacted if process technology ‘Y’ has significantly lower manufacturing yields than anticipated?” In a scenario process, decision makers invent and then consider, in depth, several varied stories of plausible futures for their impact on the long-range plan and strategic capacity requirements. Hence, scenarios are “scripts” about the way the world might turn out tomorrow, scripts that can aid in the recognition and adaptation to changing aspects of the present environment that could affect the future balance between capacity needs and capacity availability [23].

1.1.3 Strategic Capacity Planning at Intel Corporation

The strategic capacity planning function at Intel Corporation is an integral element in the long-range planning process. Each quarter the Intel capacity plan for all fabrication, sort, assembly, and test facilities over the 5-year horizon, and the demand forecasts for the full range of Intel’s logic and flash memory products, are reviewed as part of the long range planning (LRP) process. The chief objective of this planning effort from a high level is to assess the degree of difference (gap) between prior quarter’s planned production capacities and the current quarter’s predicted capacity requirements, over the LRP horizon, and to make information-based decisions on changes to the roadmap. Figure 1 represents this general process flow.

The quarterly process begins, as shown in Figure 1, with revised component demand forecasts from the various Intel divisions. One such component family is Intel’s flagship Pentium 4 ® processor. In addition, forecasts for the manufacturing parameters are gathered in order to determine the quantity and timing of silicon wafers needed to meet a given unit demand. This demand requirement is expressed in wafer starts per week (WSPW). The gap between predicted wafer starts per week requirements and the current capacity plan (also

expressed in wafer starts per week) represents the primary driving force of the LRP decision-making process. If these decisions are made without a coherent guiding philosophy or rigorous data analysis and discussion, the future “structure” of the firm could be compromised. Consequently, the LRP process is buttressed with an interdependent array of systems, procedures, and decision-making forums designed to address this tension in the plan and to make resource allocation decisions estimated to be of maximum benefit to the firm.

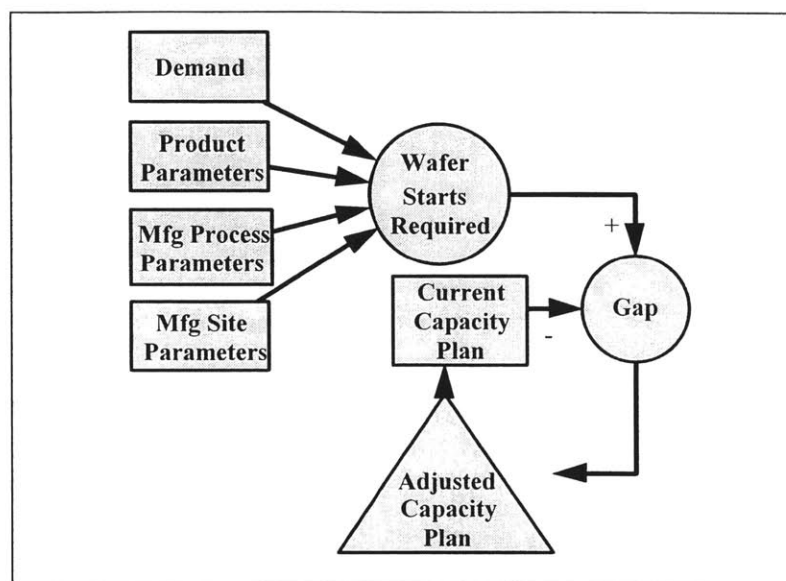


Figure 1: LRP Process

The end result of the LRP process is a published roadmap that details the factory network’s production plan by component, fabrication/sort facility, process technology, assembly package platform, and assembly/test site. Investment decisions are then executed based on the revised roadmap, completing the LRP process loop for the respective quarter. The quarterly review frequency enables quick response to variation in demand forecasts, technology readiness, manufacturing parameters, and factory capacity. The five-year horizon facilitates planning for a variety of long-lead time requirements, including technology development, factory space positioning, headcount deployment, equipment procurement, and supplier/customer requirements. Indeed, equipment lead times can range from a few months to over two years, while new factory construction can take several years from plan to factory startup. To understand the importance of this planning process to Intel, one need only review

the relative amount of investment Intel makes in capital purchases annually, as exhibited in Figure 2.

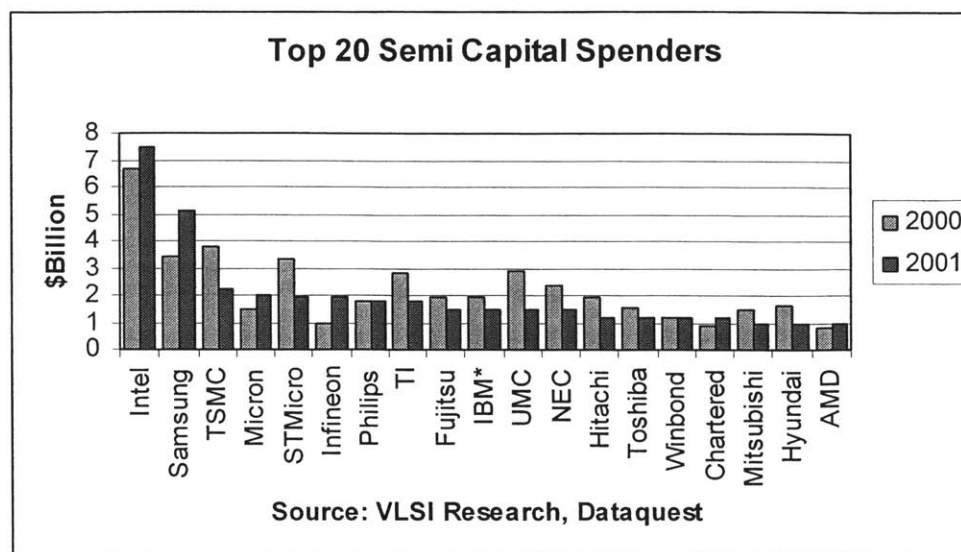


Figure 2: Semiconductor Capital Spending

A central strategic capacity-planning (SCP) group drives the capacity planning process within the organization. This group is the conduit through which demand information from the product divisions and manufacturing parameters from the various factories are translated into WSPW requirements. The primary responsibilities of SCP are to coordinate the LRP process, provide analytical and decision-support expertise (including what-if analyses), generate graphical representations of data to aid decision makers, manage the central LRP database, and to publish and document the LRP. Hence, SCP does not make the capacity decisions, rather influences them through its documented procedures and methodologies. For a more comprehensive discussion of the SCP team structure and business process, refer to Page [19].

1.2 MOTIVATION FOR THIS WORK

The effectiveness of the LRP process depends not only on the quality of the data (e.g. demand forecasts, parameters forecasts, etc.), but also on the depth and clarity of the data exploration and presentation. Mountains of data will serve no purpose unless the decision-makers engage in activities to promote a “sense making” of the data. What-if scenario

analysis and visual depictions of information are two key approaches used by the SCP group to accomplish this end. The what-if scenarios generated as part of the “qualitative” analysis of the strategic capacity planning process require that data be accessed, manipulated, and transformed into some sort of visual representation of the data – for instance, a temporal graph. The numerous steps required to transform the data into information has the effect of limiting the range of scenarios explored. Similarly, the need to effectively convey detailed information to a decision-making forum suggests that improved display methodologies would prove beneficial. Hence, there exists an opportunity to develop alternative information visualization techniques. The underlying hypothesis is that fresh visual tools can be used to communicate an idea, as well as to discover the idea itself.

This work is inspired by the author’s belief that alternative visualization tools can be used in the strategic capacity planning process to improve the “information work”, or the activities of foraging for data, thinking about data, and acting on data. The purpose of visualization is insight, not pictures. The main goals of this insight are discovery, decision-making, and explanation. Information visualization is useful to the extent that it improves our ability to perform these and other cognitive activities [6].

Finally, the scope of this effort is aligned with the vision of the SCP team, as evidenced in two of its management objectives [25]:

“Develop leading edge software solutions that enable high quality information analysis”

“Provide information, not just data, to stakeholders”

1.3 APPROACH AND METHODOLOGY

The goal of this project has been to develop a set of alternative computer visualization tools to aid SCP analysts and their stakeholders in the exploration and presentation of data. Two distinct areas are identified for information visualization tool deployment: the what-if analysis process, and the visual presentation of detailed, hierarchical data.

1.3.1 What-If Visualization Tool

The bulk of data exploration is contained within the what-if scenario analyses, tens of which are completed each quarter. As the base case forecasts are point estimates, what-if

analyses are a critical component of the qualitative forecasting approach. These analyses require the use of a non-graphical relational database tool for translating the product parameters into WSPW requirements, as described in Section 1.1.3. Data is then exported to Excel¹ for graphical presentation. This has the effect of limiting the range of scenarios explored.

The idea of an interactive, dynamic, and easy-to-use software tool enabling flexible interaction to this circuitous process has considerable appeal. The system could be powerful enough to be of real benefit to the user, yet require a minimum amount of the user's time to specify the conditions and interpret the answer. If such a system is feasible, then it seems reasonable that it might have some utility.

Thus, the first goal of this effort was to see if it is possible to create a functional, visual display system. An interactive visualization tool has been developed to facilitate the extraction of a data set from the central database, and the manipulation of the data using dynamic sliders. This tool uses Visual Basic for Applications in an Excel environment, thus providing for a high degree of automation, while leveraging the spreadsheet and graphical capabilities of Excel. This tool is not a simulation model in the sense that it does not involve random variables (as with Monte Carlo simulation) or complex feedback processes (as with system dynamics). Rather, it is a simple deterministic model that emulates the formulaic structure of the WSPW database. Hence, the user can expect to receive results similar to those of the WSPW engine.

1.3.2 Presenting Detailed, Hierarchical Data with Treemaps

Equally important is the ability to effectively convey the long-range plan information to decision makers. Given the hundreds of components, tens of processes, and tens of factories, this information is by nature both hierarchical and dense. For instance, one may wish to view the WSPW requirements for each product in the entire factory network, organized by factory or division. Such a display could lead to instant insights regarding product routing strategies. This would be difficult to accomplish using standard graphical methodologies. Treemaps, a recent display innovation, efficiently exhibit both data attributes and data hierarchy in a mosaic-type two-dimensional area. These plots have the ability to reveal hierarchy and high-

¹ Excel is a registered trademark of Microsoft Corporation.

detail complexity in a single view. Treemap 3.0¹, a Java application developed by the University of Maryland, is used in this research effort to depict various data sources from the long-range plan. Figure 3, for example, conveys WSPW information for a collection of products organized by factory. The area of the respective rectangles represents the contribution of each component relative to the total WSPW requirements. Consequently, the secondary thrust of this thesis is to obtain some evidence on where in the capacity planning process such a display medium might be used, and on what classes of problems the tool could help solve.

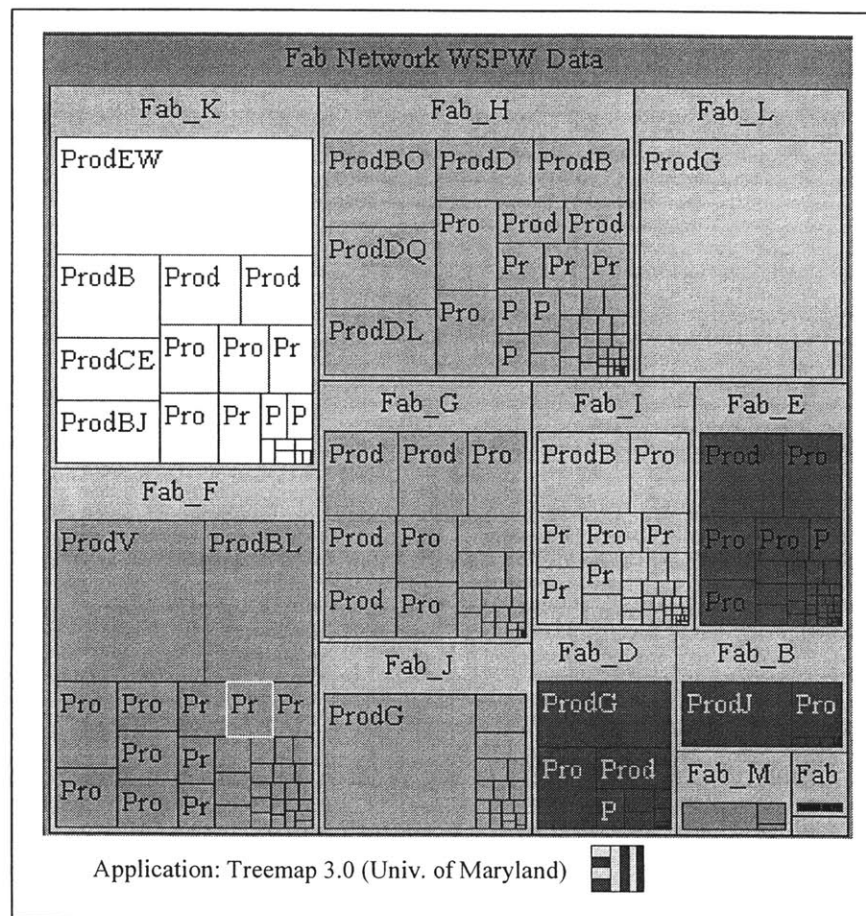


Figure 3: Treemap Display - Product Data Sized by WSPW and Organized by Fab

¹ <http://www.cs.umd.edu/hcil/treemap3>

1.4 THESIS STRUCTURE

This thesis is organized into six chapters.

Chapter One argues the case that effective strategic capacity planning must be a core competency of capital-intensive semiconductor firms such as Intel Corporation. The capacity planning problem is introduced, and a proposal for the inclusion of two unique information visualization tools is offered.

Chapter Two considers the emerging field of information visualization from a historical and theoretical perspective, introducing models and guidelines for the effective deployment of such tools. Additionally, this chapter provides background information for the treemap data display technique.

Chapter Three introduces the What-If Visualization Tool (WVT) in the context of the frameworks of the second chapter. This chapter discusses the WVT in detail, including the choice of model platform, the tool architecture, structure, mathematical underpinnings, and user interface components.

Chapter Four reveals some of the key features of Treemap 3.0 from a usability perspective, and work done to improve the utility of the tool vis-à-vis the human interaction model.

Chapter Five presents results and analysis of the What-If Visualization and Treemap display tools. This chapter discusses some of the assumptions and approximations used within the WVT, and highlights the results obtained both from the perspective of numerical accuracy and usability. Moreover, this chapter illustrates the representative use of treemap displays in the decision-making process.

Chapter Six highlights some key observations and learnings acquired from the approach taken in this thesis, and suggests recommendations for future work.

2 INFORMATION VISUALIZATION AS A FIELD OF STUDY

Effective information visualization techniques are becoming increasingly important in a world paradoxically characterized by abundance of data and scarcity of decision-making time. In order to elucidate this field of study, the emergence and subsequent evolution of information visualization are introduced. The advantages of information visualization – namely, as a means to improve cognition and to transform data into information – are disclosed using lead practitioner models and constructs. Usability considerations are then illustrated as an essential set of guidelines for information visualization tool design. Finally, the treemap display methodology is presented, focusing on the motivation, capabilities, and general underpinnings of the treemap display technique.

2.1 ORIGINS OF INFORMATION VISUALIZATION

Information visualization is a burgeoning field of study that is perhaps best described by its cognoscenti. Ware defines information visualization as a graphical representation of data or concepts, and the process of constructing a visual image in the mind [36]. Spence posits that information visualization is the process of forming a mental model of data, thereby gaining insight into that data [28]. Finally, Card, Mackinlay, and Shneiderman define information visualization as the use of computer-supported, interactive, visual representations of data to amplify cognition – cognition being the acquisition or use of knowledge [6].

Using the first two definitions in the prior paragraph, one would certainly conclude that the field of information visualization has been in practice for quite some time. In fact, one could argue that this field is rooted in the original capabilities of man to express oneself via paintings and other sketches. Yet according to Tufte, it was not until the late 1700's that statistical graphics – time-series, scatter plots, and multivariate displays to name a few – were invented, long after such mathematical triumphs as calculus and probability theory [31]. William Playfair (1759-1823), a Scottish political economist, seems to have been among the earliest to use abstract visual properties such as line and area to represent data visually. The first known time-series using economic data was published in his book, “The Commercial and Political Atlas” [31]. One could argue that just as scientific inventions such as the telescope

and microscope have enabled us to see things in new ways, visual representations such as maps and time-series plots have also enabled us to view things differently.

The definition proposed by Card et al. differs from the prior two in that it specifically uses the modifier “computer-supported”. Indeed, the 1990s represent perhaps the genesis of the modern field of information visualization, as the power and ubiquity of the personal computer enabled the creation and use of powerful tools for data retrieval and display. This suggests that information visualization has been an active research topic for just over a decade [16]. In a sense, the confluence of innovative computer technologies and the explosive rise in available data fueled the need for improved visualization tools. Indeed, a major challenge facing software designers is organizing and presenting the huge masses of data generated from many sources. As Sterman suggests, data presentation and visualization will remain a critical arena for future software development [30].

2.2 VISUALIZATION AS A MEANS TO IMPROVE COGNITION

A basic tenet of information visualization is that it allows one to gain a more complete, rapid understanding of the underlying data that is being explored or presented. Mackinlay states that visualization is about using computer graphics to think about more cases, more variables, and more relations [16]. Card et al. state that information visualization is useful to the extent that it enhances our ability to engage in the activities of discovery, decision-making, and explanation [6]. Certainly, it is possible to design systems based on the aural or tactile senses; however, vision by far has the greatest bandwidth to process information. The power of visualization then is its ability to amplify cognition, creating an alternative, high-speed conduit for deriving new knowledge and insights about the world.

Card et al. propose six prime ways in which visualizations amplify cognition [6]. These are listed below and elaborated upon in subsequent paragraphs.

1. Increasing users’ memory and processing resources
2. Reducing the search effort
3. Using visual representations to support pattern detection
4. Enabling the construction of perceptual inferences
5. Improving the perceptual monitoring process

6. Allowing for manipulation of the display medium

1. Increasing users' memory and processing resources

One could argue that visualization techniques could increase human memory and processing resources by transferring memory storage requirements from the user to the visualization tool. Working memory, also called short-term memory, is the temporary buffer where we hold concepts, recent percepts, and plans for action [36]. Working memory is very limited, with a decay rate for visual stimuli on the order of 200 milliseconds [7]. A user scanning through a data table for instance may need to identify or “mark” data elements in order to later translate the data into information. Visual representations can shift the burden from the user by providing visual markers (e.g. ticks on a line graph) and by revealing structural relationships among the components (e.g. a map) [36].

Another salient benefit derived from data visualization is the sheer quantity of information that can be presented and interpreted. A topographical map, for instance, can convey distance, relative spatial arrangement, and contour – all within a two-dimensional space. Think of a regular map, and how it enables much better route planning than would be possible with a set of poorly remembered facts about cities and their relative locations [36]. Techniques have also been developed to display highly dimensional, structured data within a single view. The old adage that a “picture is worth a thousand words” is apropos, as a user would need, for example, to activate a greater amount of working memory to study a data table versus a graphical representation of the data.

2. Reducing the search effort

Data grouping, data organization and data retrieval can enhance the search effort. A stacked column graph, for instance, conveys both aggregate levels as well as the relative constituent values. Range bars can be added to line graphs to give a sense for the variability around the expected values. Color and texture can be used to bring the attention of the user to a particular object or area of the screen. A common data grouping technique is the Venn diagram, which shows interrelatedness and hierarchy of information. Data retrievals could be enhanced, for example, through user interfaces. In typical database retrievals, the user must apply two types of knowledge: knowledge of the mechanics of searching (e.g. query syntax),

and the knowledge of the conceptual aspects (e.g. data tables and their relationships) [7]. A potential enhancement is to use a “menu” selection or some other graphical interface as a front end to the database.

3. Using visual representations to support pattern detection

Recognizing and recalling patterns in a visual display is much simpler than recalling actual data. For example, when asked to draw a sine wave from memory, one may indeed not know the values for each integral degree value from 0 to 360 degrees. However, one would probably recall the general oscillatory shape of a sine wave, as well as some typical values such that the sine of zero degree is zero, and the sine of ninety degrees is one. From that information alone, the individual could construct a reasonable approximation to an actual sine wave. This example demonstrates memory recall of visual patterns. Separately, we could ask how one perceives or detects patterns in the first place.

Ware refers to a set of Gestalt “laws” which may shed some light on how we detect patterns [36]. Some of these laws are listed in bullet form below:

- Proximity: The tendency to group objects in close proximity together.
- Similarity: The tendency to group similar objects together.
- Closure: The tendency to view a closed contour as an object
- Continuity: The tendency to construct visual entities out of smooth, continuous elements

These laws perhaps can explain the effectiveness of the Venn diagram. In a Venn diagram, we interpret the region within a closed contour as defining a set of elements. Multiple closed contours are used to delineate the overlapping relationships among the different sets [36]. The perceptual factors of closure and continuity can help explain the effectiveness of this diagram.

4. Enabling the construction of perceptual inferences

Visualizations can aid in the rapid formation of inferences from the data. The basic boxplot exemplifies the ability of an abstract object to effectively convey information about

the underlying data. As an illustration, the boxplot in Figure 4 below provides the following information about the data set it comprises [35]:

- The center of the data set
- The range of the majority of the data
- The spread of the unquestionably “good” data
- Possible outliers

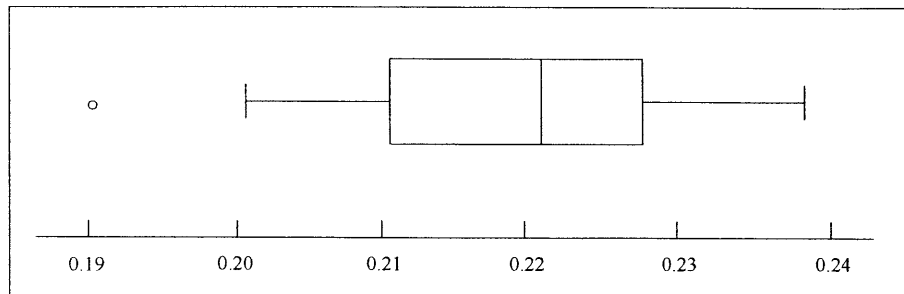


Figure 4: The Basic Boxplot¹

The boxplot above allows one to gauge the relative centering and spread of the data set. The median, represented as the line drawn within the box, represents the middle value of the data set – it is a measure of “centering” of the data. The outer sides of the box represent the inner 50% of the data set – 25% on either side of the median. The vertical lines marking the end of the “whiskers” represent the “inner fences”, or adjacents. The adjacents represent the most extreme data values considered unquestionably good from a statistical perspective. Thus, so far we have displayed the typical value (the median), the range of the “bulk” of the data (the box defined by the quartiles), and the bounds for the statistically good data (the whiskers) [35]. Finally, the small circle represents an outlier – a data point that falls beyond the limits for statistically “good” data. In sum, the boxplot facilitates a quick understanding of the relative nature of the data set.

5. Improving the perceptual monitoring process

Perceptual monitoring describes the notion of a user responding to events displayed in a graphic visualization. Perceptual monitoring techniques are integral to supervisory control

¹ For a more comprehensive discussion on boxplots, refer to Vining [35].

systems. Supervisory control is a term used for complex semiautonomous systems that are only indirectly controlled by human operators [36]. Examples include aircraft and power station monitoring systems. A key goal of supervisory control systems is to create effective ways for a computer to gain the attention of a human, using motion, color, notification messages, and even sound. It is equally important to use information visualization techniques to convey the pattern of events, as can be exhibited through the use of temporal graphs. One such example is the run chart used to track the behavior of a parameter (typically a quality parameter) over time.

6. Allowing for manipulation of the display medium

A mere rearrangement of how the data is displayed can lead to a surprising degree of additional insight. As we often do not know what to expect of the data when graphing, we frequently find ourselves formulating new questions about the data [28]. Following inspection of a graph, a user may then decide upon further rearrangement or the addition or removal of certain data. Bertin states:

“A graphic is no longer ‘drawn’ once and for all: it is ‘constructed’ and reconstructed (manipulated) until all the relationships which lie within it have been perceived.” [3]

Pivot tables, often found in spreadsheet programs like Excel, allow the user to “pivot” the dimensions of the table – for example, to transpose column headings to row positions or vice versa. Brushing is an interactive technique that enables subsets of the data elements to be interactively displayed in a “pop-up” mode. Panning is the smooth movement of a viewing frame over a two-dimensional image of greater size, while zooming is the increasing magnification of a decreasing fraction (or vice versa) of a two-dimensional image given a viewing frame of constant size [28]. These techniques and others can be used to allow patterns in the data to reveal themselves.

2.3 TRANSFORMING DATA INTO VISUAL INFORMATION

Card et al. state that visual aids serve two related though distinct purposes: (1) communicating an idea, and (2) using graphical means to discover the idea itself [6]. An expansion of this model is the idea of using information visualization to improve information

work in general, which can be thought of as three main activities: (1) foraging for data, (2) thinking about data, and (3) acting on data [16]. Indeed, Card et al. describe visualization as the “mapping of data to visual form that supports human interaction in a workspace for visual sense making” [6]. As discussed above, interactivity is not a requisite for effective visual representations. Spence affirms this viewpoint in stating that the design of visualizations is a craft activity, the success of which depends on the designer’s understanding of the task for which the tool is intended [28]. Nevertheless, interactivity can offer enormous value in visualization design. The human interaction model proposed by Card et al. and exhibited in Figure 5 presents a good framework for discussion.

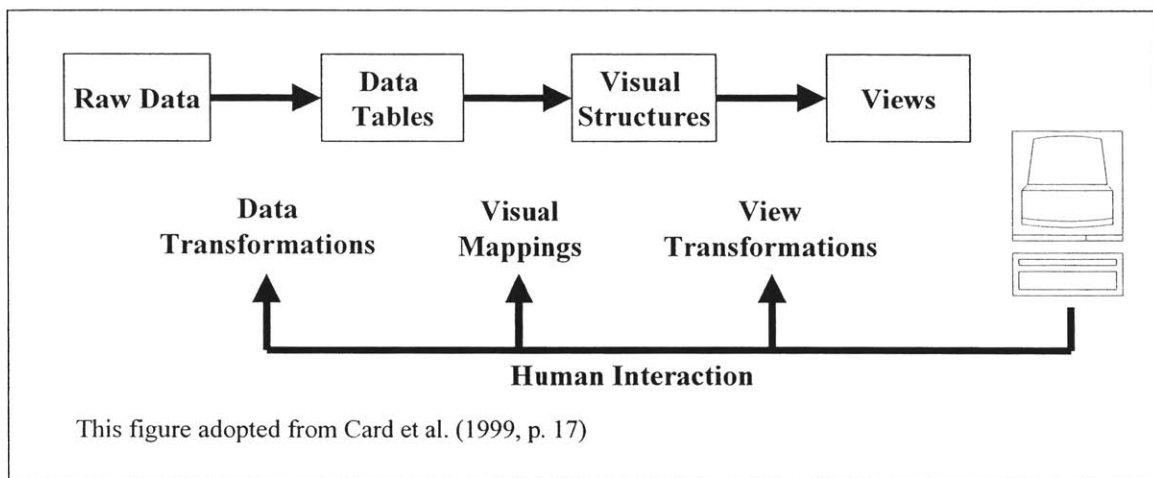


Figure 5: Human Interaction Model

The human interaction model depicts the flow of raw data into data tables, visual structures, and views. In many ways, this is analogous to the sequential concept of transforming data first into information then into knowledge. Where passive visualization displays rely solely on pattern recognition to do the task, active controls involve the user in developing hypotheses and refining them through an augmented visualization process. The purpose of visualization is to elicit from the user a response of the form “ah-ha” indicating that sudden insight has been obtained to some effect, insight that could lead to enhanced understanding or a new idea [28].

Raw data is typically transformed into some sort of structure, a relational data set for instance, in order to facilitate the creation of visual structures. The goal of a data warehouse for example is to structure information for aggregation and preprocessing. Robertson,

Mackinlay, and Card state that almost all information to be displayed on a computer can be classified as unstructured, linear, or hierarchical [21]. An example of a linear structure for data is a temporal line graph, while hierarchical data includes such things as file structures or organizational charts. In a relational database, data is stored in tables, which are composed of rows and columns. Tables of independent data can be linked, or “related”, to one another through common “keys”. The issue with relational databases is that they require an understanding of data structure and query syntax. Hence, a separate user interface can be employed, such as a graphic-oriented environment where the user can connect the inputs and outputs of objects representing the data. Additionally, Shneiderman demonstrates the use of dynamic queries as a visual alternative to SQL for querying databases [6].

Proceeding from data tables, or structured data, to visual structures is the essence of creating visualizations. The user can interact through the visual mapping process. Often data must be selected prior to display. The pivot table discussed above is a good example as it allows the user to click and drag variables for display. In other cases, data may need to be transformed. Interactive data mapping is the process of adjusting the function that maps the data variables to display variables. The interaction may impose a simple scaling function, or a non-linear transformation such as logarithmic or square root. Figure 6 depicts a simple “slider” tool for performing such a transformation function [36]. Finally, view transformations enable the user to manipulate the display medium. As described in Section 2.2 above, panning, zooming, and brushing are some of the techniques employed that empower the user to “walk” through the visualization.

In conclusion, Ware argues that one way of describing the effectiveness of an information visualization tool is the length of time needed to acquire useful information [36]. The cost of knowledge depends on the utility of the user interface, which may include a visualization coupled with a user interface. Given a responsive, well-designed tool, people can obtain information far more rapidly and accurately than otherwise. Thus far, we have discussed the benefits to be derived from information visualization tools. The following section briefly highlights some important concepts to be considered when designing a graphical user interface.

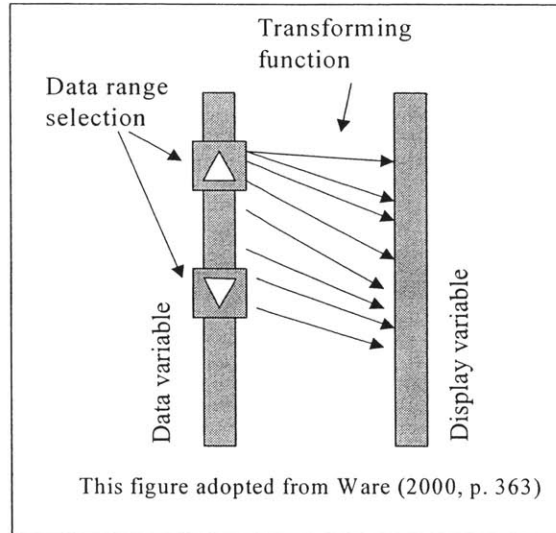


Figure 6: Interactive Mapping

2.4 USABILITY CONSIDERATIONS

Thus far, we have discussed the potential of visualization techniques to increase the “cognitive power” of individuals to transfer data into information and knowledge. As these visualizations are typically generated on a computer, it is critical that the user interface be designed correctly. This is especially important when designing information visualization tools that are either interactive, allowing the user to manipulate the display, or dynamic, where continuous movement of a control causes continuous rearrangement. If the interface is not properly designed, there is a risk that the visualization tool can pose a hindrance, or even worse, lead the user toward incorrect decisions and actions. The intent of this section is not to provide a detailed analysis of user interface design, but to bring attention to this topic as a critical element in information visualization design, and to briefly discuss a few of the general concepts contained therein. These concepts illustrated below are (1) metaphors, (2) S-R compatibility, and (3) affordances and constraints.

The idea behind using metaphors in interface design is to invoke something familiar and typically concrete to help users understand the functioning of a particular interface. Computer interfaces incorporate many metaphors in order to make the computer more usable [7]. One of the most prevalent and successful metaphors has been the desktop metaphor in computer operating systems. We are all familiar with how to manipulate the items on a desktop, using the mouse or keyboard as a “pen”, and selecting one’s files from the “file cabinets”, typically

icons that resemble a folder for instance. Other interface metaphors gaining increasing acceptance include objects that resemble buttons, scrollbars, and menu displays. Eberts states that the main use of the metaphor is to get novices accustomed to the system so that they can use it, interact with it, and learn more about how it works [7]. In general, the interface should be designed to enhance exploratory behavior, and include logic to ensure that errors are either preventable or recoverable. An extension of this idea is to strive for what Rutkowski refers to as the “principal of transparency” [36]. When transparency is achieved, the user is able to immerse oneself in the visualization task. The tools themselves seem to disappear. Thus, metaphors can aid the user in forming accurate mental models of the data being visualized.

Stimulus and response (S-R) compatibility refers to the effectiveness of a display in conveying the connection between spatially separate objects. Eberts illustrates this concept with a familiar example [7]. Consider a stovetop, complete with four burners and their respective control knobs. Figure 7 depicts two configurations, one with a good S-R compatibility (A), and the other with a poor S-R compatibility (B).

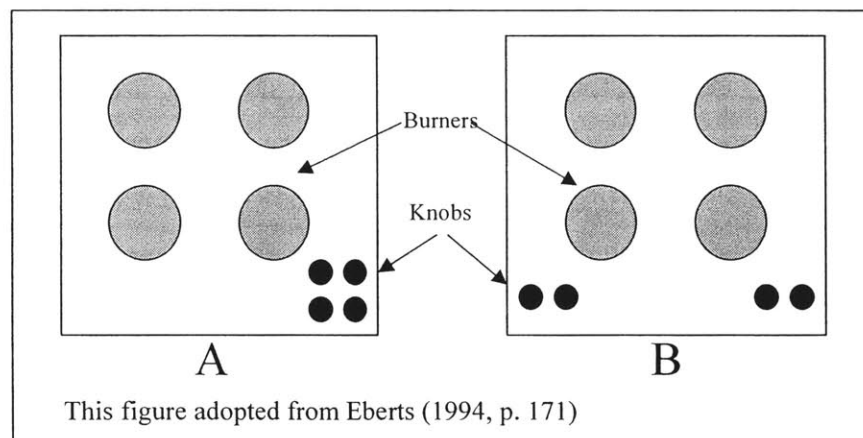


Figure 7: S-R Compatibility Example – Stovetop Configurations

In A, both the burners and knobs are laid out in a square. It is easy to deduce that the top left knob corresponds to the top left burner and so forth. This configuration represents high S-R compatibility. In B, on the other hand, the knobs are laid out in a line from left to right. Though it is easy to infer that the left pair corresponds to the left two burners, it is not clear how the two knobs relate to the front and rear burners. Labeling would be required to ensure accuracy. Consequently, configuration A would be less error prone than B, and would require

less time for the user to process the response given the stimulus [7]. S-R compatibility plays an important role in creating effective user interfaces for information visualization.

Affordances and constraints refer to particular characteristics of objects that appeal to human perception. Generally, affordances “invite” the user to perform an action, while constraints “limit” the possibilities for action. Gibson describes affordances as the “perceived possibilities for action” of an object [36]. Norman, in his book “The Psychology of Everyday Things”, defines affordances as “the perceived and actual properties of the thing, primarily those fundamental properties that determine just how the thing could possibly be used” [18]. For instance, we perceive surfaces for walking, knobs for turning, handles for pulling, space for navigating, tools for manipulating, and so on. Thus, to design a good user interface, we must use appropriate affordances to simplify the user’s task. Two examples of such affordances in computer interfaces are buttons and scroll bars. Buttons are typically depicted in such a manner that they appear to have a raised surface, so that, like buttons in the real world, they can be depressed (using a mouse click as opposed to a finger). The scroll bar object contains a “box” and directional arrows on either end of a bar. The box in the scroll bar indicates something that can either be grabbed or moved using the arrow objects [7].

Constraints are the antithesis of affordances. Norman states that the world naturally restricts the allowed behavior of objects, such as the way objects can be manipulated [18]. One such example is the modern electrical plug, which is typically designed with one larger blade so that the polarity of the plugged connection is consistent. One popular computer interface constraint is the menu display [7]. With the menu display, the user does not type in a value but must select from a predetermined list. This eliminates misspelling errors among others. Another example is the concept of “graying out” objects or text that are not functional at a particular time. Finally, the interface can be designed to query the user upon certain actions such that the user has to verify they wish to perform a certain action.

In sum, affordances can be used in an interface to increase the accessibility and ease of use, while constraints can be used to reduce certain types of errors [7]. Nevertheless, these techniques, for all their benefits, can lead to increased screen clutter or reduced flexibility. Thus, a designer must use a balanced approach, always considering the intended goal of the application.

2.5 TREEMAPS: A DATA VISUALIZATION METHODOLOGY

2.5.1 Treemap Overview

Treemaps comprise a class of visualization techniques that was pioneered in the early 1990's by Ben Shneiderman at the University of Maryland. Shneiderman was motivated by what seemed at the time a latent need – the ability to display the directory tree structure of a computer's hard disk in graphical form. The central idea was to display the structure as well as the file size so that one could easily infer where disk space was being consumed [27]. Figure 8 represents both the listing structure from using the well-known DOS command <TREE> as well as the tree diagram representation, using an arbitrary directory.

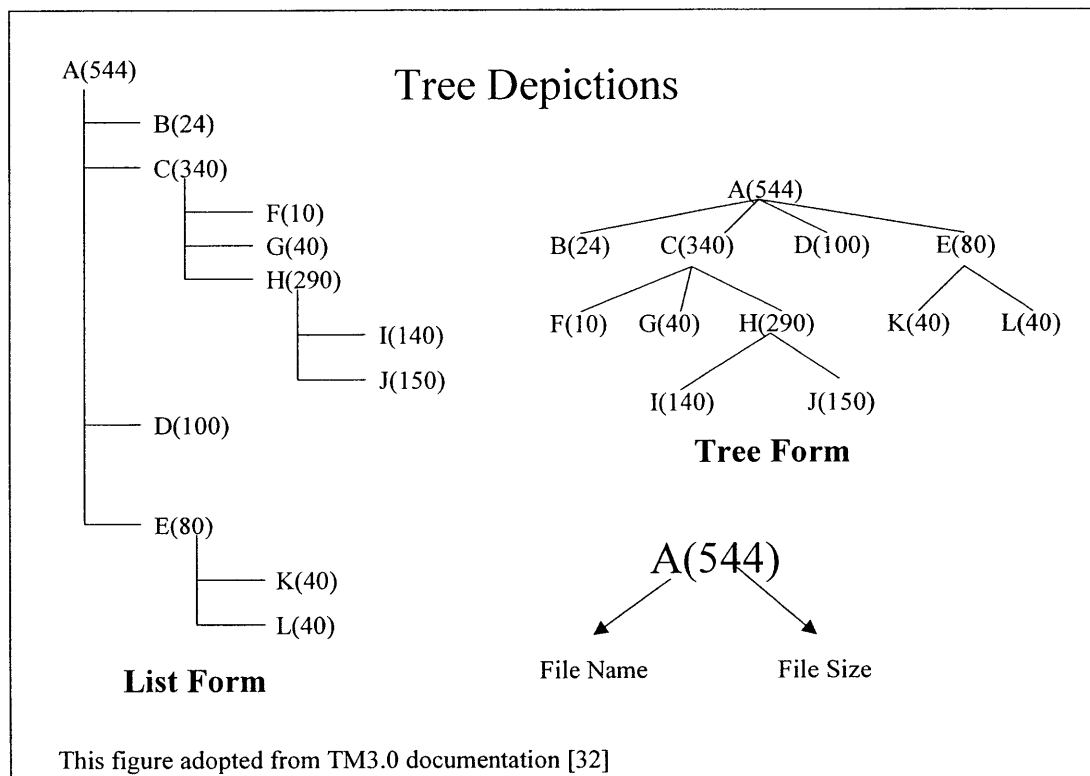


Figure 8: Comparison of List and Tree Structures in Hierarchical Data Display

The advantages of the tree form diagram are immediately obvious. As the diagram exhibits, the number of display lines needed to represent the list form hierarchy is directly proportional to the number of nodes in the hierarchy. In this instance, 12 lines are required to display files A through L. The tree form method displays the same information in 4 lines, and

is arguably more effective in conveying the structure of the data. Nevertheless, the tree diagrams are themselves limited.

The two prime limitations of the tree diagrams are (1) their inefficient use of display space and (2) the lack of content information. Though superior to the list form, a typical tree diagram makes use of only 50% of the available display space, the remaining space being used as the background or “white space” [6]. For a small data set such as that in Figure 8, this does not pose a problem. However, for large trees this traditional node and link approach would be limited by the available display space. Another issue with tree diagrams is their use of textual information – in the case of Figure 8, file size. As Shneiderman notes, global views of large tree diagrams require the nodes to be rendered so small as to obscure the visual cues designating node content [6]. These deficiencies prompted the development of a novel space-filling approach that maps the full hierarchy onto a rectangular region. Hence, the term “treemap” describes the notion of turning a tree display into a two-dimensional planar map [27].

A visualization of a treemap representation of the structure in Figure 8 would increase our understanding of the definition proposed above. This is revealed in Figure 9, using a Java application developed by the University of Maryland.

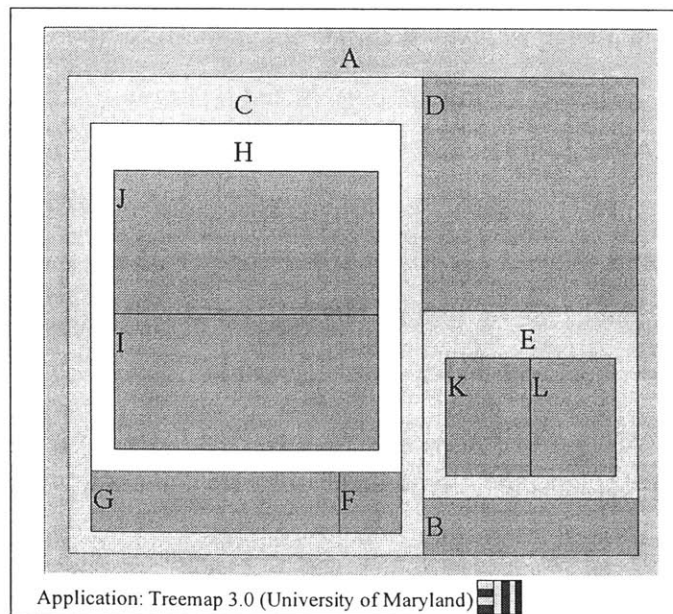


Figure 9: Treemap Representation

The treemap above depicts both the structure and the size of the attributes displayed. For instance, note that elements B, C, D, and E belong to parent A. Subdividing further, we note that K and L comprise E, while F, G, and H make up C. Finally, H is comprised of J and I. The frames, or shaded and labeled enclosures, can be viewed as file folders, while the self-contained rectangles represent the individual files. The frames are used to delineate the sub-node structure through containment, while the size of the rectangles represent, in this instance, file size. Stated another way, the treemap is comprised of a number of closed-contour regions within a larger closed contour – thus representing conceptual containment. What immediately stand out are the relative size of folder C as a whole, as well as the individual files D, J, and I.

The procedure is simple. Beginning with a designated root node one draws a rectangle – usually, in order to make effective use of the screen, it will be almost the same size as the display. Within the parent rectangle are smaller rectangles, one for each subordinate node of the node under consideration [28]. For instance, in Figure 9, files J and I are subordinates of folder H, which in itself is a subordinate of folder C, and so on. This construction is repeated until all nodes are drawn. Once derived, individual rectangles can be encoded by color to denote some attribute, such as file size or days since creation [28].

Color can be used to code information or to display an additional attribute of the rectangular objects. For example, color may be used for labeling an object as in nominal information coding. When we wish to make it simple for someone to classify visual objects into separate categories for instance, giving the objects distinctive colors is often the best solution [36]. Alternately, color could be used to represent an additional variable. A color range could be selected to represent a separate variable, where the saturation or color contrast depicts the size of the variable. New-York-based Smartmoney.com has incorporated a treemap display tool where stocks are represented by colored rectangles, and color represents financial indicators such as the percent increase (green scale) or decrease (red scale) of the particular stock for that day [37].

In sum, treemaps represent both the hierarchical structure and qualitative information contained within a data set using a two-dimensional rectangular space. The example in Figure 9, however, only contains a handful of data elements. The real power of treemaps, in the author's opinion, is their ability to represent a large amount of data in a relatively small space, or its high data-density capabilities. Hence, a key enabling technology of treemaps is its

underlying algorithm for laying out the rectangles. The squarification algorithm is perhaps the most effective method for such displays.

2.5.2 Display of High-Density Data

Tree drawing algorithms have been geared toward efficient and aesthetic methods to display objects representing the data. The slice and dice algorithm of the original treemap used parallel lines to divide a rectangle representing an item into smaller rectangles representing its children. For each level of the hierarchy, the parallel lines were alternated between vertical and horizontal [26]. The issue with the slice and dice approach is that as the data set grows, or as the relative size of particular rectangles becomes very small, there is a tendency for the rectangles to become razor thin. The net effect is that these rectangular data elements become difficult to identify, label, and compare in size [26].

Alternative layout mechanisms have been developed to cope with the drawbacks of the original slice and dice approach. Perhaps the most widely proliferated layout algorithm in use is the aptly named “squarification” approach. The aim of the squarification algorithm is to create low aspect ratio rectangles as opposed to thin elongated rectangles, where aspect ratio is defined as the following:

$$\max(\text{height} / \text{width}, \text{width} / \text{height}) \quad (1)$$

Some salient advantages of the low-aspect ratios are the following [4]:

- Square items are easier to identify, label, and point at (i.e., with a mouse pointer)
- Comparison of the size of rectangles is easier when their aspect ratios are similar
- Display space is used more efficiently. That is, the proportion of the border width to rectangular width/length is better for the low-aspect ratio rectangles versus the elongated.

Figure 10 demonstrates the visual superiority of the treemap display versus a typical display – a pie chart in this instance. The pie chart representation is analogous to the slice and dice approach; only here we use a circular region in lieu of a rectangular region.

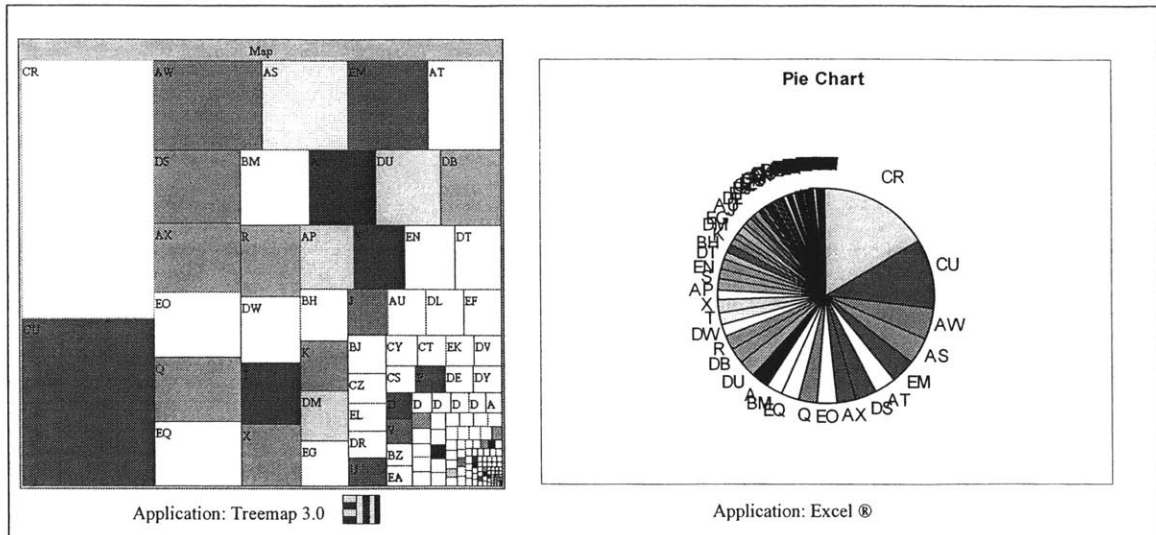


Figure 10: Treemap Display versus Pie Chart - 156 Data Elements Depicted

Both the treemap and the pie chart were created using an identical, non-hierarchical data set containing 156 elements. For clarity of presentation, the data range for the pie chart was sorted in descending order, while the squarification algorithm performs its own sorting sequence. The advantages of the treemap as outlined in bullet format above are readily observable. The squarification algorithm utilized in the treemap display yields rectangles that are easy to identify and label, while the pie slices are too thin to be labeled, and quickly become indistinguishable. The squarified elements can also be selected more effectively in a computing environment where the treemap application allows mouse-over display, view navigation, zooming, or other enhanced functionalities. Finally, the treemap display affords relative size comparisons and data organization due to an apparent arrangement of elements in descending order from top-left to bottom-right. As the squarification algorithm is a key enabler of the treemap display methodology, the next section draws upon a simple example in order to illustrate its generic functioning.

2.5.3 Squarification Algorithm

The treemap is constructed via recursive subdivision of the initial rectangle into sub nodes (typically frames) and finally into the “leaf” attributes. Initially, treemaps were subdivided in such a manner that thin, elongated rectangles were the result [4]. This shortcoming eventually led to the concept of squarification algorithms – so called because of

their aim to draw rectangles with low aspect ratios, the ideal value being one (note: a square has an aspect ratio of one). Though squarification algorithms may vary, the following simple example, the idea of which is adopted from a paper by Bruls et al., demonstrates the general concept [4].

The general squarification methodology for a simple, one-level data set can be summarized as follows:

1. Determine the width and height of the parent rectangle.
2. Sort the “leaf” rectangles in descending order by area.
3. Split the parent rectangle vertically or horizontally, depending on whether it is wider than it is tall or vice versa.
4. Add the first “leaf” rectangle into this first half. Calculate its aspect ratio (i.e. $\max(\text{height}/\text{width}, \text{width}/\text{height})$).
5. Add the next “leaf” rectangle beneath or beside the first, depending on the initial subdivision (vertical or horizontal respectively).
6. Compute the aspect ratio of the newly added rectangle and compare to that of the previous rectangle. If the aspect ratio has improved (i.e. is lower), then leave the rectangle as drawn and repeat Step 5 with the next rectangle. If the aspect ratio has increased, then “lock-in” the prior rectangular arrangement, and repeat the process beginning with Step 3 for the current rectangle.

This heuristic method generally yields a geometric arrangement typified by rectangles sorted in decreasing size from the top left to the bottom right of the display space. This natural flow from top left to bottom right assists the user in identifying both the large and small rectangles. An example will elucidate this sequential process.

Example

Suppose we have a parent rectangle of width 7 and height 5 (total area = 35), and wish to subdivide the rectangle into six smaller rectangles of areas 12, 8, 8, 4, 2 and 1. Using the sequential process detailed above, we could draw the rectangles as shown in Figure 11.

First, we make a vertical split in the parent rectangle, as it is wider (7 units) than tall (5 units), and then draw the 12-unit rectangle. The aspect ratio of this rectangle is 2.08. Next, in

step 2 we draw the next rectangle beneath the first. The aspect ratio of this rectangle is 2, which is an improvement over the original (step 1) arrangement. We try again with the third rectangle, only to discover that the aspect ratio for this rectangle has increased to 3.94. Therefore, we decide that we have attained a local optimum for the left section in Step 2, and begin processing the right half in step 4 [4].

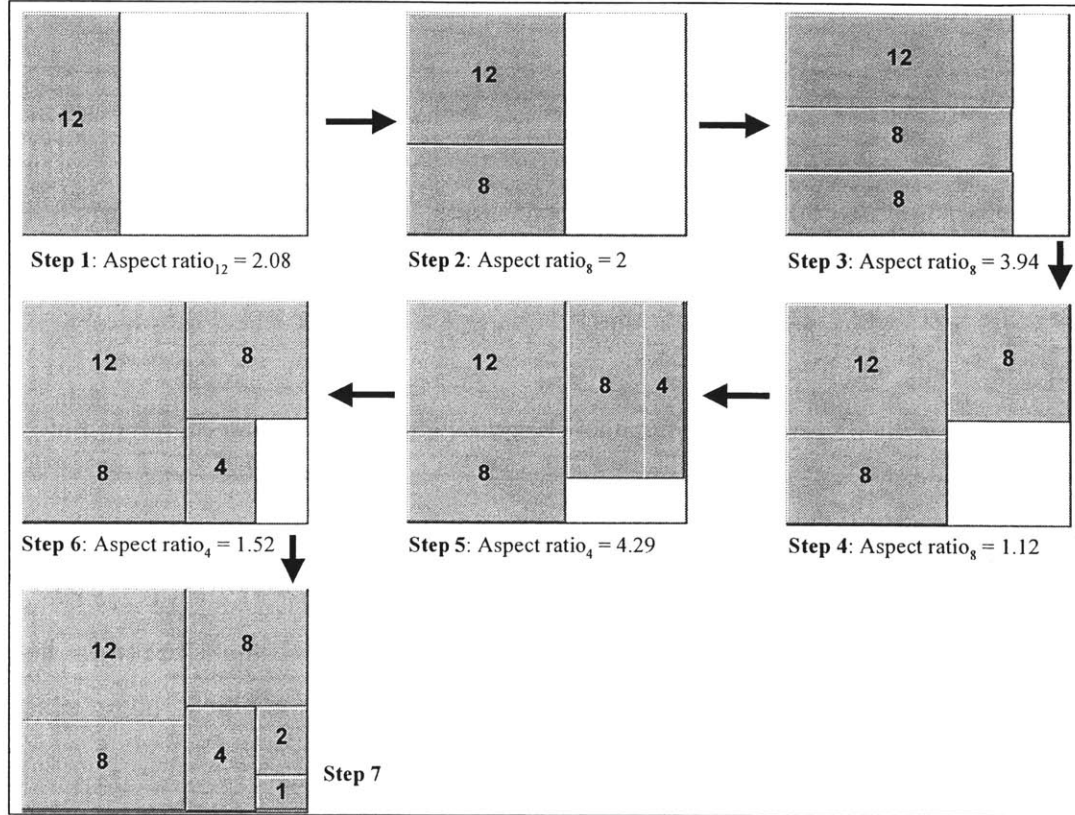


Figure 11: Squarification Algorithm Sequence

We make a horizontal subdivision in step 4, as the remaining, unfilled rectangle is taller than it is wide. The aspect ratio of the third, 8-unit rectangle is 1.12. When, in step 5, we add another adjacent rectangle, the aspect ratio of this added rectangle is 4.29. Hence, like before, we lock in the configuration of step 4 and subdivide the remaining unfilled rectangle using a vertical division, and so on. Step 7 yields the final result, a collection of “squarified” rectangles. As one can see, the rectangles so displayed are easy to identify and compare. Though various squarification algorithms can and have been used, the preceding example illustrates a simple technique that obtains locally optimal aspect ratios. The end result is a rectangle layout that facilitates data identification and comparison.

3 THE WHAT-IF VISUALIZATION TOOL

This chapter provides a comprehensive discussion of the What-If Visualization Tool (WVT). An empirical analysis is presented to expose the prevalent type of scenario analyses studied within SCP. The existing process for conducting scenario analyses is then introduced, using the human interaction model described in Chapter 2. The limitations of this process are considered, and a new conceptual framework – that of the WVT – is offered. The aggregate semiconductor supply chain is brought to the fore, along with the mathematical framework that is the core of WSPW calculations. Initial efforts to craft the tool in a system dynamics modeling environment are discussed. The software architecture of the functional WVT is subsequently presented, along with a discussion of its relative benefits vis-à-vis software interoperability. Finally, the structure and features of the WVT are depicted.

3.1 FRAMEWORK FOR THE WHAT-IF VISUALIZATION TOOL

The what-if analyses quantify the impact of various scenarios on capacity requirements in wafer starts per week (WSPW), and are a critical component of the “qualitative” aspect of strategic capacity planning. The purpose of the what-if analysis is to explore alternative “images” of the future, as opposed to using trending or causal models to “predict” the future. In a scenario process, managers typically invent and then consider the ramifications of several varied stories for the future. Within Intel, SCP management ratifies and quantifies these scenarios, with input provided by the product divisions. These images provide a vehicle for understanding how plausible outcomes could affect capacity requirements, as opposed to extrapolating prior trends. This is not to suggest that trending and causal modeling are inferior, but to assert the need for a dual approach to strategic capacity planning.

Transforming the various scenarios into capacity (in WSPW) requirements is accomplished through the LRP Database. The LRP Database is the capacity assessment tool containing the minimum granularity and quantity of data required for strategic capacity planning of semiconductor components. It serves both as a data repository for forecasted product, process, and factory parameters and as a calculation engine for converting unit product demand into WSPW requirements. Parameters include unit demand, line yield,

throughput time, and die¹ size. Typically twenty to thirty what-if scenarios are ratified each quarter for quantitative analysis via the LRP Database. Generic scenario descriptions typical of the semiconductor industry are provided in Table 1.

What If Description		Impact
1	Overall market size for Family A projected to increase by 5%.	Demand
2	Product A_2 will be delayed two quarters – pull in Product A_3.	Demand
3	Process N will experience a steeper learning curve during ramp.	Die yield
4	Product G will be canceled - pull in Product H to substitute.	Demand
5	Shift production of Product E from Process N to Process N + 1.	Die size
6	Product C die size forecasted to be 10% larger.	Die size

Table 1: Sample What-If Scenario Descriptions

As is illustrated in Table 1, each scenario tells a unique “story”. Some scenarios appear to be driven more by external factors (e.g. market size increase in item 1), while others may be the result of internal strategies and tactics (e.g. decision to cancel Product G in item 4). For this research effort, the various scenarios have been categorized based on the primary WSPW conversion factor that they affect. As shown in Table 1, the what-if scenarios are typically “demand” related. Demand is increased, decreased, and/or shifted in time. A more complete empirical analysis has been conducted using the list of scenarios for four recent, consecutive quarters. This data, exhibited in Figure 12, reinforces the conclusion that, though several parameters are required for converting unit demand into WSPW requirements, historically a set of three are modulated during what-if analyses – unit demand, die size, and die yield – the overwhelming majority being unit demand related. This finding would have implications for the design of the What-If Visualization Tool (WVT).

3.1.1 Limitations of the Current What-If Process

The current what-if analysis process has been evaluated using the human interaction framework outlined in Figure 5. This framework provides guidance as to the ability of a system to support the “information work” of foraging for data, thinking about data, and acting

¹ Die: one individual device produced on a wafer. After wafer fabrication, the die are “cut” from the wafer.

on data as described in Section 2.3. The particular procedure for conducting what-if work is depicted in Figure 13 and discussed below.

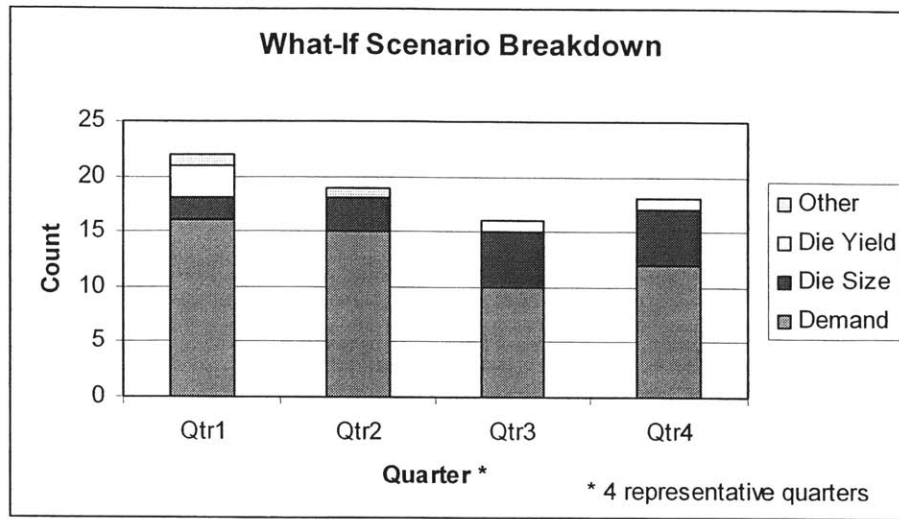


Figure 12: What-If Scenario Categorical Analysis

The first step involved in a typical what-if analysis is to create a distinct database file by copying and renaming a base case as the what-if case. The chief reason for this approach is that such analyses characteristically involve the modification of a relatively small portion of the total information. Hence, beginning with a base case and subsequently modifying is in the interest of both speed and error avoidance.

The next step involves the actual editing of information stored in the what-if database. The LRP Database as a whole is a client-server system comprised of a Microsoft SQL¹ *back-end* component (the database server), which is used to store and retrieve data and protect the databases, and a *front-end* component (the client), which is used to present and manipulate data [13]. The edits, represented as the “outline data changes” activity in the figure, reflect the assumptions of the particular what-if under consideration. For example, a demand change scenario would involve editing the demand fields for the product(s) under consideration, where each field represents a particular quarter in the LRP horizon. Division input, SCP team knowledge, and peripheral software tools are used to formulate the exact “script” of the what-

¹ SQL: a client/server database management system. SQL is an acronym for Structured Query Language.

if scenario. Finally, note that the data editing described above completes the data transformation phase of the human interaction model.

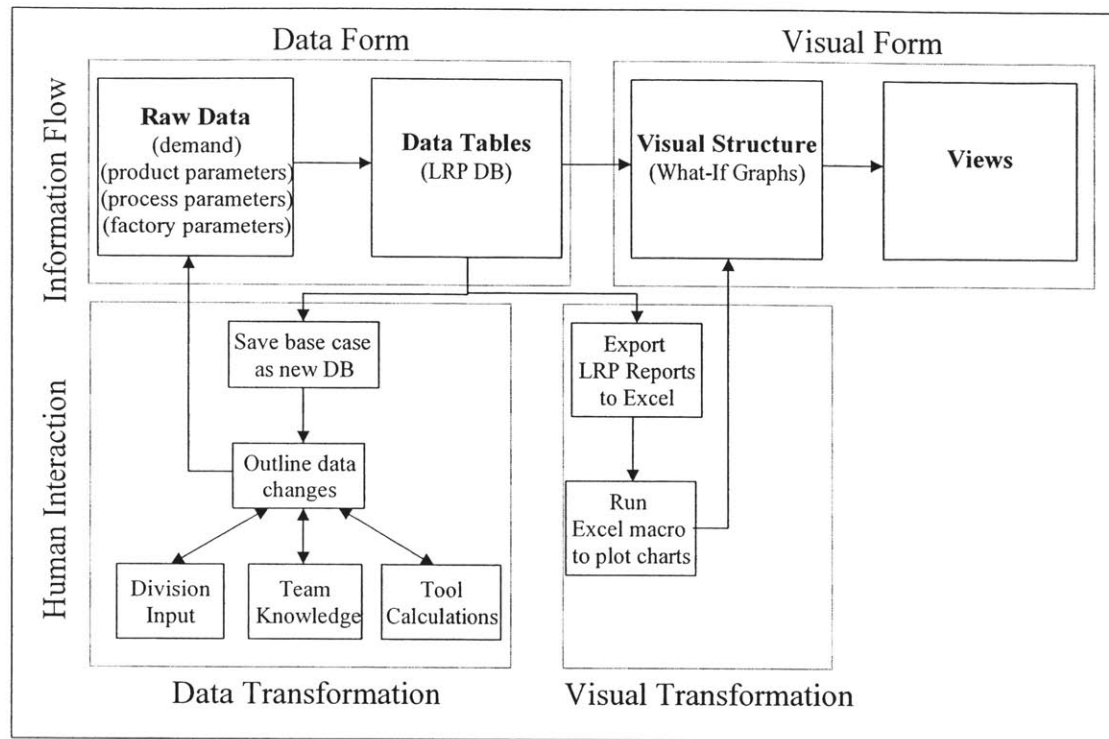


Figure 13: Current What-If Process - Human Interaction Framework

The data presented within the LRP Database is in tabular form. This data must be represented graphically in order to maximize the conceptualization of the data by decision makers. This transformation is represented as the “visual structures” element in the information flow process. As is shown in Figure 13, the data stored in the database is exported to Excel, and then graphed using a macro¹ designed specifically for this task. The resulting temporal graph is typically a pair of line graphs depicting the WSPW requirements (vertical axis) for each quarter in the LRP horizon (horizontal axis) for both the “base” case and the “what-if” case. Finally, the “views” element simply represents the ability of the user to manipulate the view (e.g. zooming) in order to study the data.

The what-if process as described above represents a sequential, indirect process for manipulating data and viewing that data in graphical form. As a change is suggested, it

¹ Macro: a sequence of instructions that the computer carries out when order to do so. In this context, the macro was created in the Visual Basic for Applications (VBA) environment.

requires re-working the database and regenerating graphs to view the effects. Typically, what-if scenarios require from two to eight hours to complete, depending on the task. Furthermore, as explained in Section 3.1, the majority of scenarios historically have not utilized the vast amount of data available in the databases. As a result, it has been judged that an alternative visualization tool could be designed so as to provide a convenient, powerful adjunct to the what-if scenario process.

3.1.2 Concept for the What-If Visualization Tool (WVT)

The vision of the WVT is to enhance the current what-if process by allowing the user to interactively engage in exploratory data analysis through a real-time, graphical environment. Manipulation of the factors that impact WSPW requirements, particularly demand, die size, and die yield, would lend the power of rearrangement to the user, who could ask “what-if” questions and receive immediate feedback. Thus, in addition to communicating an idea, the graphs could be used to discover the idea itself [6]. Following inspection of the graph, the user may then decide upon further rearrangement or exploration of an alternative scenario. As Spence postulates, the “very property of interactivity can considerably enhance the tool’s effectiveness, especially if the computer’s response follows a user’s action virtually immediately” [28]. Figure 14 depicts the value of such a tool using the human interaction framework.

Drawing a comparison from Figure 13, one deduces that the WVT functions both to modify the data and to graphically depict the data. In effect, it serves to “short circuit” the process by providing a direct path from the raw data into visual structures. The WVT portrays the state of variables contained within a particular database (i.e. the base case) chosen by the user. Based on the empirical analysis depicted in Figure 12, the variables available for manipulation are unit demand, die size, and die yield. For the sake of manageability and user needs, the author posits that this restriction would not limit the utility of the model. Users manipulate the variables graphically by selecting a particular quarter (horizontal axis) for a given variable, and modulating the value of the variable using a dynamic slider. The effect is seen both on the variable display and on the total WSPW. Indeed, as the WSPW graph must morph to reflect the change in the variable, the WVT must

contain the identical logic and data used by the LRP Database in calculating WSPW requirements.

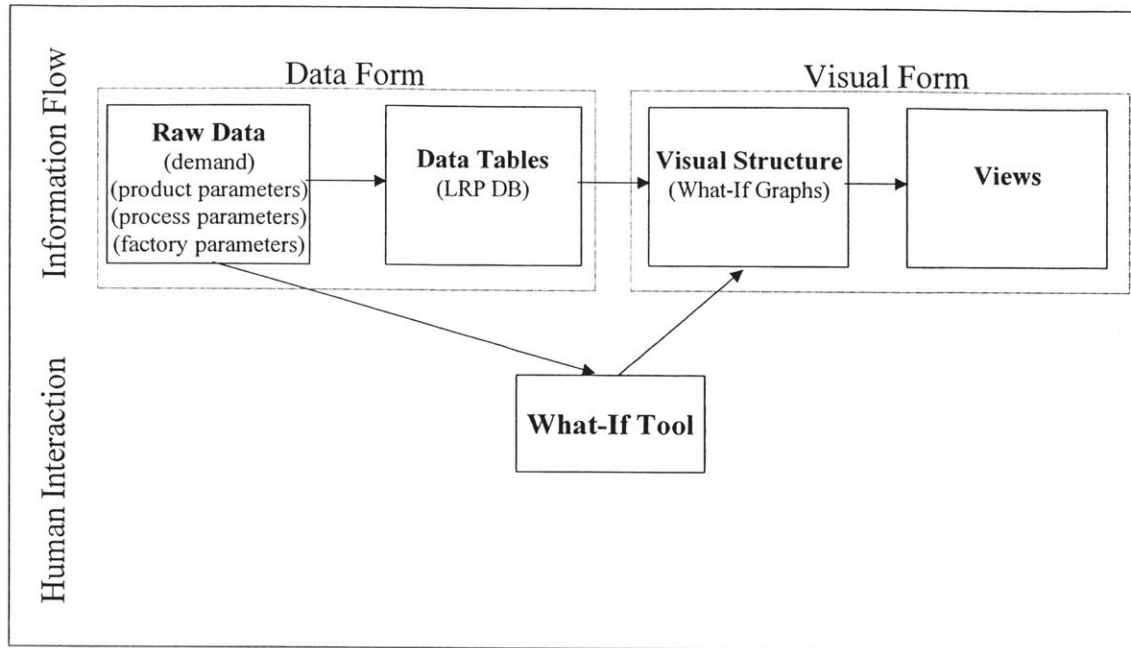


Figure 14: What-If Visualization Tool - Human Interaction Framework

The calculations performed by WVT are transparent to the user. Though there is nothing physically direct about using a mouse to drag a slider on the screen, the user obtains the illusion of direct control given the rapid and compatible feedback of the WVT [36]. The system portrays the state of the critical variables. The graphs of these variables, altered to reflect new strategies or hypotheses, could in turn suggest other changes [24]. Hence, the WVT could serve as a medium for stimulating thought and in understanding the relations between critical variables and WSPW requirements.

The WVT is not intended to supplant the existing what-if scenario process. Indeed, the current process allows for detailed modification of complex data. For instance, a change in total product family demand would require the reworking of numerous data sheets in the LRP Database. As the WVT tool hinges on single component selection prior to manipulation, the tool is inadequate for quantifying the impact of such a scenario. Nevertheless, it is believed that the WVT would be a useful complement to the existing tools – most notably, for classes of problems related to high-volume components that drive capacity. For such occurrences, the WVT would have the following benefits:

- Reduce the time required for viewing (in graphical form) the impact of a scenario on WSPW requirements.
- Encourage increased scenario exploration and refinement.
- Facilitate the comprehension of non-linear relationships between critical variables and WSPW requirements.
- Provide a means to view the calculations used to transform unit demand into WSPW requirements.

Given the increasing complexity of product markets, technological development, and manufacturing in the semiconductor industry, these benefits could be of real value to the planning organization.

3.2 WSPW CALCULATION MODEL

The What-If Visualization Tool calculates wafer starts per week using database parameters filtered by user-specified product/process combinations. The underlying analytical structure of the WVT mirrors that of the LRP Database. The intent of this section is to give the reader a general idea of the aggregate supply chain model for semiconductor manufacturing, including the underlying analytical framework for calculating wafer starts backwards from a desired unit demand.

3.2.1 Semiconductor Manufacturing Supply Chain Overview

In this section, we introduce a generic structure for the semiconductor supply chain and a simple mathematical model for calculating wafer starts per week. As introducing inventory points makes the mechanics of WSPW calculations more complex, we build the intuition in two sections.

Semiconductor wafers are processed through fabrication, sort, assembly, and test stages before being shipped to regional distribution centers as shown in Figure 15. This is important from the perspective of WSPW planning, as the respective yield losses and throughput times for each stage need to be accounted for when planning the quantities and timing of wafer starts required to support an expected consumer demand. The ensuing discussion thus seeks

to illuminate some of the basic manufacturing concepts associated with each macro process stage, as well as to formulate a set of general equations for calculating wafer starts to support a given demand.

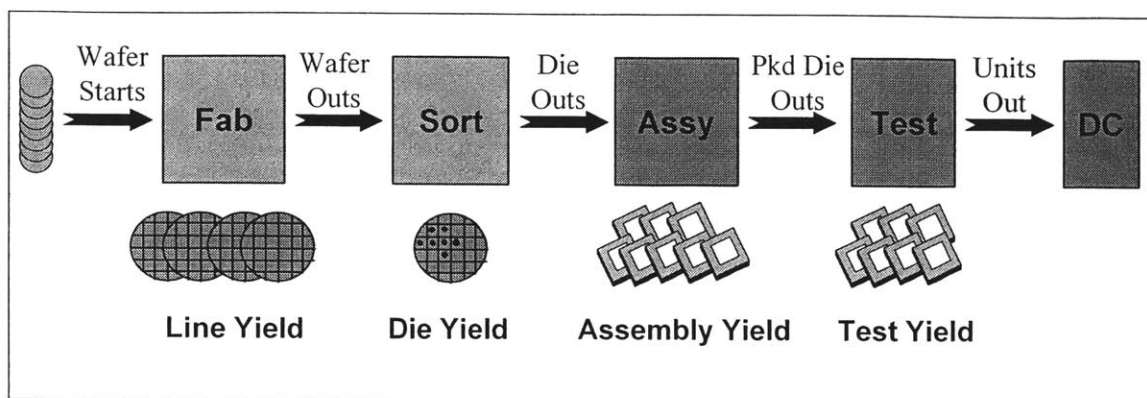


Figure 15: Semiconductor Manufacturing - Throughput time and Yield Effects

Fabrication

Semiconductor processing begins with raw silicon wafers, typically doped¹ to certain specifications. These wafers are then processed within capital intensive, complex fabrication facilities, referred to as “fabs”. Within a fab, typically hundreds of individual die are created in a grid-like fashion over most of the wafer’s surface. Each individual die contains millions of electrical components such as transistors, resistors, and capacitors, specifically interconnected to form a functional semiconductor device such as a microprocessor or a memory chip.

The manufacturing process flow is essentially divided into a “front end” sequence, in which individual components are fabricated within and above the silicon substrate, and a “back end” sequence, where multiple levels of patterned insulating and conducting materials are added to interconnect the individual components within each device. The process is also subdivided into various modules, including patterning (photolithography), material removal (etching), material layering (vapor deposition and sputtering), and doping (thermal diffusion and ion implantation). The individual wafer will cycle back through these various operations many times, totaling several hundred manufacturing steps. Among these, patterning is the

¹ Doping refers to the process of embedding impurities into “pure” silicon to alter its electrical properties.

most critical. Indeed, a generation of process technology is usually defined by the chip's minimum feature size, or "linewidth", which is the minimum dimension that can be defined on the chip (e.g. 90 nanometers). This linewidth is typically gated by the resolution of the lithographic process.

Finally, the processing sequence so described typically results in an aggregate throughput time on the order of several weeks, with accompanying yield losses a natural byproduct of even the most finely tuned manufacturing process. Fab yield can be represented with the following simple formula:

$$Fab_Yield(FY) = \frac{Wafers_Out}{Wafers_In} \quad (2)$$

Sort

Following the wafer fabrication is wafer sort, also known as die sort or electrical sort. These series of tests are the report card of the manufacturing process. In sort, each chip is tested for electrical performance and circuit functioning [33]. The goal of testing is threefold: (1) to identify working chips before they proceed to the packaging operation; (2) to characterize the device along performance parameters (e.g., operating speed of microprocessors); and (3) to determine the actual yield of working chips, which will be used to resolve discrepancies between actual and forecasted die yield and to provide feedback to manufacturing. Die yield in semiconductor manufacturing is highly dependent on process maturity, as depicted in Figure 16. This dramatic improvement is often the result of process learnings and innovations in the launch of a new product or process.

Die yield forecasting unlike line yield forecasting for wafers, is typically represented by a yield model. Most yield models begin with the assumption that there is a random distribution of point defects characterized by some average defect density (D), with each defect independent of the others. Given this assumption, the die yield (DY) of an integrated circuit with area (A) is the probability of having zero defects within an area A, as given by the Poisson distribution [2]:

$$Die_Yield(DY) = e^{-AD} \quad (3)$$

Finally, die yield is then coupled with the gross die (GD) available on a wafer - given a certain die size - to calculate the expected number of functional die per wafer (DPW).

$$DPW \propto GD * e^{-AD} \quad (4)$$

As Equation 4 shows, the number of good die on a wafer is a non-linear function of the defect density.

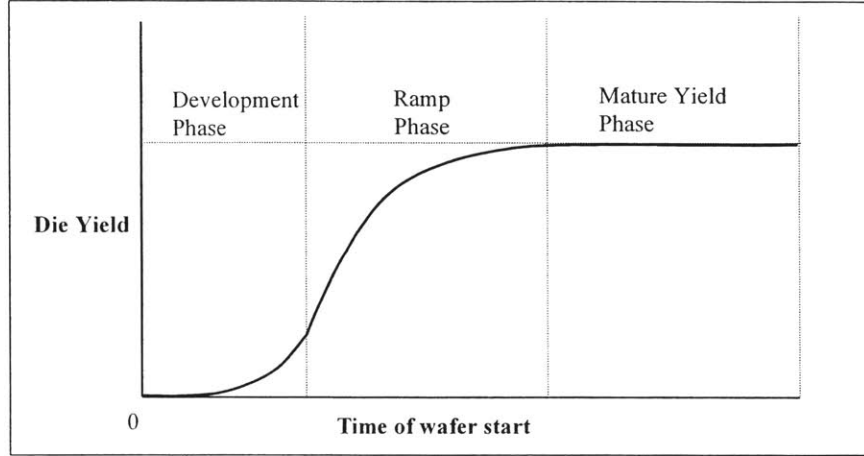


Figure 16: Die Yield as a Function of Process Maturity

Assembly and Test

At the completion of sort testing, wafers are shipped to assembly/test facilities. The wafer is first placed under a diamond saw and cut up into individual die. The bad die are discarded, and the good die are reserved for packaging. In the packaging process, the individual die are attached to an enclosure that serves both to protect the device and to provide the interconnections to the rest of the world – for instance, the motherboard. Final testing includes structural testing, electrical testing, and accelerated life testing. As with the manufacturing process, there will be some yield losses in the assembly/test processing. This is represented as the number of good units per good die:

$$A/T_Yield(ATY) = \frac{Units}{Die} \quad (5)$$

Wafer Start Calculation

The wafer start per week (WSPW) calculation, as inferred above, is largely a function of the various stage yields and the total throughput time. As with all planning decisions, time is divided into buckets, where a bucket is an interval that is used to break time and demand into discrete chunks [12]. Within the context of Intel's LRP horizon, quarterly buckets are used, where demand is considered equally distributed over the quarter, and where product, process, and factory parameters are assumed constant. This assumption requires that wafers be started in the quarter(s) prior to demand. Hence, the wafer starts must be apportioned across the quarters based on relative throughput times.

In the event throughput time is less than a full quarter, a semiconductor manufacturing firm needs to begin processing wafers in the quarter prior to demand in addition to the quarter of demand. A simple approximation for calculating WSPW for the respective quarter is to apportion the wafer starts based on relative throughput times. Hence, the functional notation for WSPW required to meet a demand in an arbitrary quarter 'n' is represented by the following functions:

$$WSPW_{n-1} = f(Demand_n, TPT_{n-1}, TPT_n, FY_{n-1}, DPW_{n-1}, ATY_n) \quad (6)$$

$$WSPW_n = f(Demand_n, TPT_{n-1}, TPT_n, FY_n, DPW_n, ATY_n) \quad (7)$$

Equations (6) and (7) represent the specific case where total throughput time (TPT) is less than a quarter. Certainly, a different set of equations is used for a TPT that is greater than one quarter. More importantly, these equations assume no inventory buffers in the supply chain. Thus, the "demand" term above is end consumer demand. However, as the next section discusses, buffering strategies complicate the formulation of demand.

3.2.2 Incorporating Inventory Strategies into the Model

The above formulations represent the simple case where there are no inventory buffering strategies along the supply chain. However, strategic inventory points are emblematic of an industry with long lead times and high variability of demand such as exist in the semiconductor manufacturing business. One such inventory point is the die bank, located

between fab/sort and assembly/test. Figure 17 represents the semiconductor aggregate supply chain with this planned inventory buffer. Semiconductor manufacturing firms leverage die banks as a strategic inventory point to maximize the responsiveness of the supply chain. Forecasting groups of packaged semiconductor components that share the same die parents can help reduce errors. Companies then build-to-stock into the die bank and configure-to-order downstream from the die bank. The net effect of such a die build strategy is to alter the demand terms of Equations 6 and 7 above.

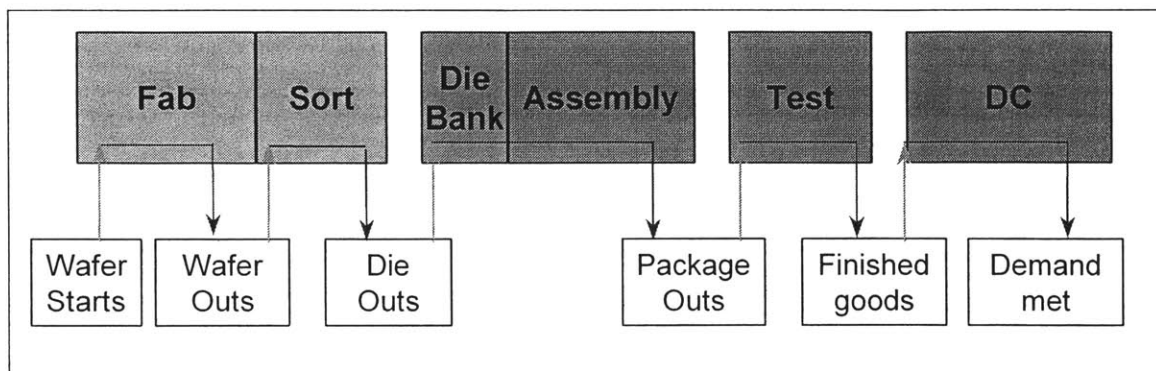


Figure 17: Semiconductor Supply Chain – Inventory Points

The planned inventory points, such as the die bank depicted above, require that die be built in advance of demand. For instance, Product A may be specified as having a requisite three weeks of inventory on hand in the regional distribution center. This stock of inventory requires that a certain amount of packaged products be built in advance. Hence, the planned build quantities for a given quarter hinge on the expected consumer demand for that given quarter, augmented by a buffer of the following quarter's expected demand, less the current stock of inventory.

The die bank represents the stock of good die, or devices, awaiting the assembly/test stages of manufacturing. Like the DC inventory, the die bank represents a strategic inventory placement point. The prime differentiator is the form of the product, as die inventory allows greater flexibility in that the unique package has not yet been mated to the semiconductor device. As the die bank precedes the DC in the supply chain, this inventory point must account for the DC build levels in addition to its own weeks of inventory strategies. Hence, the actual demand term in the WSPW formulation is determined through a recursive set of

calculations that proceeds up through the supply chain and is dependent on current inventory levels, desired inventory levels, consumer demand, and throughput times.

Summary

The aforementioned discussion is intended to illustrate the key concepts comprising the WSPW engine embedded within the LRP database, and adopted by the WVT. When inventory is included in the model, the WSPW calculations become more complex and difficult to conceptualize a priori. This complexity arises from the recursive nature of the build and beginning-on-hand relationships, as well as some of the inherent equations used. The former is driven largely by inventory aggregation and the change in the unit of measurement from packaged components to die, while the latter is emblematic of the non-linear equations involved, such as those used to calculate gross die (GD) from a given die size, and die per wafer (DPW) for a given GD and die yield (DY).

3.3 WVT EARLY DESIGN DECISIONS: INTERACTIVE VENSIM

One of the early technical decisions was whether to build the interactive tool in a simulation or an analytical environment. An interesting area of simulation where dynamic display is gaining prominence is system dynamics. System dynamics is a simulation modeling technique that deals with complex systems, particularly where feedback mechanisms and delays exist in the process to be modeled. System dynamics is composed of flowcharts coupled with equations that describe how the various elements of the system interact [29]. Hence, system dynamics modeling tools such as Vensim ® and Powersim ® are explicit and transparent, and use graphical objects to ease inspection and verification. Analytical environments, such as spreadsheet applications like Excel, are based on formulaic structure (as opposed to graphical objects) and are particularly useful for direct computations. However, spreadsheet models may not be intuitive without a solid understanding of the underlying relationships. As a result, we initially opted to exploit the dynamic display and object-based programming structure of Vensim. However, as explained below, we experienced problems achieving system performance in the simulation environment. This prompted the creation of the WVT in an Excel-based platform as described in Section 3.4.

3.3.1 Interactive WSPW Model Using Vensim 5.0

Vensim 5.0 ®, developed by Ventana Systems, is an integrated environment for building and running simulation models [34]. The transparent, object-oriented structure is one of the greatest advantages of Vensim models. Moreover, version 5.0 introduces an advanced feature for real-time interactivity of simulation models called SyntheSim [34]. With SyntheSim, one can view the model outputs during experimentation with the model variables in real time. This is facilitated by the automatic creation of “sliders” for all model constants in Vensim.

An interactive WSPW model has been developed using Vensim 5.0. This model incorporates the analytical model from the LRP Database in calculating WSPW requirements for a given vector of independent variables. Based on the what-if empirical results discussed in Section 3.1.1, the model is designed to allow for manipulation of demand, die yield, and die size using sliders as shown in Figure 18. The constants, denoted as sliders in Figure 18, are used to transform the base case variables through user specification of scalar start, duration, and base (i.e., amplitude). The resulting effect on WSPW requirements is exhibited in the line graph in real time.

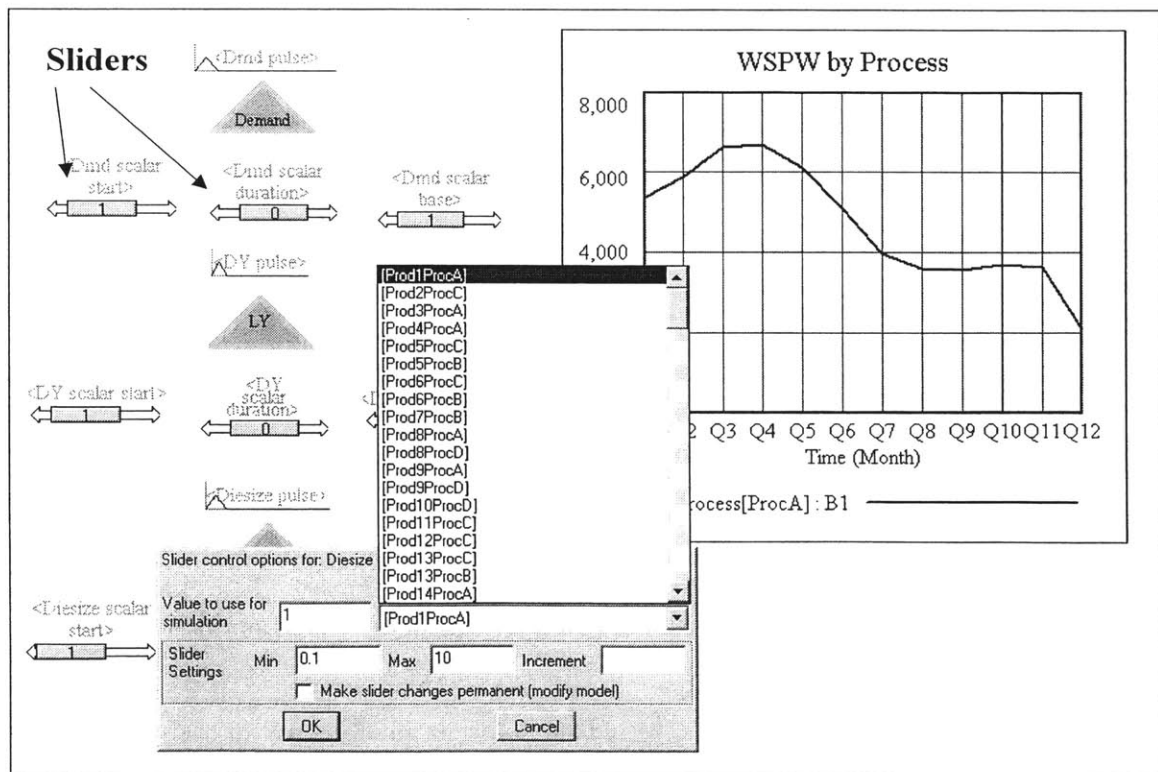


Figure 18: WSPW Model in Vensim 5.0 – Control Panel

The Vensim model comprises a set of “views” which enable the model user to easily inspect the underlying model structure. While Figure 18 represents the control panel for user interaction, Figure 19 depicts the object-oriented model structure for a subset of the demand calculations contained within the model.

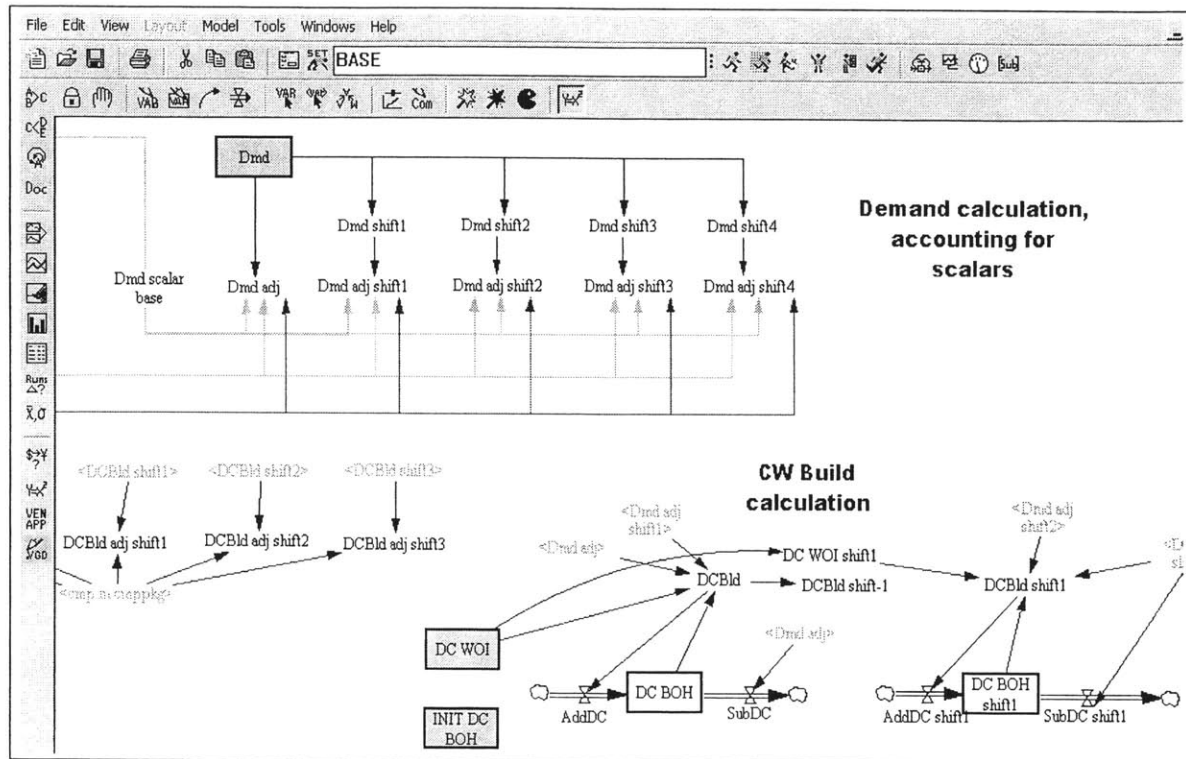


Figure 19: WSPW Model in Vensim 5.0 – Model Structure

The variables in the model sketch are represented as text, with arrows indicating dependencies. The shaded boxes represent exogenous data inputs, as in the case of external demand (i.e., Dmd), while the unshaded boxes represent stock variables, as with DC BOH (distribution center beginning-on-hand inventory). The underlying mathematical equations, though not shown, are readily available for user display and edit.

Arrayed Variables

The variables depicted in the model are vector arrays. For instance, “product demand” (Dmd) represents an array or an $n \times m$ matrix of values, where n represents the product name (e.g., Prod1) and m represents the time bucket (e.g., Q5). Hence, there are as many values for n and m as there are products in the data set and quarters in the planning horizon. Certain

variables have more than two-dimensional indices, such as “die size”. As shown in the drop-down menu list in Figure 18, die size is indexed by product, process, and time bucket. This variable naming convention simplifies the model structure and provides for concise flow diagrams. The construction of these arrayed variables is enabled through Vensim’s subscripting language, which manages the tracking and storage of these vector arrays [34].

Data Connectivity and Vensim Simulation

The interactive WSPW model replicates the analytical structure of the LRP Database. The stock and flow nature of the system dynamic variables is consistent with the recursive structure of the LRP Database calculations. In simulation mode, Vensim steps through each time increment as specified in the model, performing the requisite calculations and displaying the simulation results. External data, retrieved from a formatted Excel spreadsheet and converted to a Vensim data file (binary file), are used to populate the exogenous variables prior to simulation based on the user specifications. As the process of creating the data file requires several steps, it is necessary to furnish the data file prior to interactive simulation with SyntheSim. Consequently, it is necessary to create a data set sufficiently large to allow the user to engage in effective scenario analysis.

3.3.2 Data Proliferation and Effect on Model Interactivity

As mentioned above, the number of steps required to create a Vensim data file from information stored in the LRP Database dictates the need for a large set of data to fuel the model. The user then selects the particular product/process/factory combination from a list of values stored in the dataset (represented as subscripted variables), and subsequently engages in what-if exploration using the scalar sliders as shown in Figure 18. However, the interactive response of the model is found to deteriorate as much from the SyntheSim design as from the sheer number of variables present in the model. With SyntheSim, Vensim instantly re-simulates the entire model with each slider movement. Consequently, the entire model is recalculated for any movement of a slider. Moreover, the subscripted array structure of Vensim in some cases generates variables that are not present in the data set. Though innovative techniques have been employed to streamline the variables, the interactive response has not sufficiently improved to the point of “transparency” as posited in Chapter 2.

Subscripted Arrays and Variable Expansion

The subscripted array structure enables Vensim to uniquely identify input variables through a set of dimensions. One such input variable used in the WSPW calculations is die size. Die size is uniquely identified by product name and manufacturing process. Product 1 on Process A for instance may have a different die size than the same product on Process N. The subscripted variable for die size given these assumptions is shown below.

$$die_size(Product, Process) \quad (8)$$

Vensim generates as many variables for “die size” as there are total combinations of “product” and “process”, independent of whether the combination exists in the data set. For instance, Table 2 below represents a small data set for an arbitrary list of five products. As there are four distinct products and three processes, the subscript structure generates 12 (4 X 3) total variables, more than twice the actual number required.

Product	Process	Q1	Q2	Q3
Prod_1	Proc_A	100	100	100
Prod_1	Prod_B	110	110	110
Prod_2	Proc_A	50	50	50
Prod_3	Proc_C	400	400	400
Prod_4	Proc_B	500	500	500

Table 2: Illustration of Subscripted Arrays - Die Size by Product/Process

This variable permutation is particularly problematic with three-dimensional arrays, as the proliferation of variables is increased by an order of magnitude. In order to effect more explicit array formation, an array mapping function of Vensim has been used.

Streamlining Arrays Using Vector Mapping

The Vector Element Map function of Vensim allows for the creation of one-dimensional array from a multidimensional array. This allows one to substitute one-dimensional vectors for multidimensional arrays, expanding the length of the vector as necessary to obtain matched vectors for analytical computation. The Vector Element Map function is of the form shown in Equation 9.

$$VECTOR_ELM_MAP(vector, offset) \quad (9)$$

We illustrate the vector element map function with a simple example. Table 2 represents a two-dimensional array (excluding “time” index) of product and process for variable “die size.” We can also represent die size as a one-dimensional array of length five with argument “Prod_Proc”. Let’s assume this die size input variable is subsequently used in a computation that includes the die yield variable, a one-dimensional vector of length 10, similarly constructed and containing the argument “Prod_Proc_Site”, indicating that it is uniquely identified by product, process, and manufacturing site. In order to expand the die size variable to a vector of length 10 and ensure one-for-one correspondence, an offset variable must be added to the data set for each element in the die size array, denoting the positioning of the values for die size in the 10 element vector. Table 3 illustrates this concept.

Die Yield	Die Size	Offset	VECELM (Die Size, Offset)
Prod1ProcASite1	Prod1ProcA = 100	0	100
Prod1ProcASite2	Prod1ProcB = 150	0	100
Prod1ProcBSite2	Prod2ProcC = 175	1	150
Prod2ProcCSite3	Prod3ProcC = 180	2	175
Prod3ProcCSite1	Prod4ProcA = 200	3	180
Prod3ProcCSite3		3	180
Prod3ProcCSite2		3	180
Prod4ProcASite1		4	200
Prod4ProcASite2		4	200
Prod4ProcASite3		4	200

Table 3: Expanding Die Size Vector Length Using Offset Variable

The “Offset” variable is formed in the data file using a macro for automatic generation given the data from the LRP Database. Thus, we add this extra variable to reduce the permutation of variables that is endemic of the subscripting process.

Model Performance and Key Learnings

The vector mapping significantly reduces the number of variables generated through the initial subscripting. Using a representative data set from the LRP Database, the vector element mapping yields 80% less variables than the subscripted model structure. Though there exists a noticeable improvement in system performance (as measured by interactive response), the extent of the improvement is less than expected or desired. This is likely the result of the vector mapping function itself, which must sequence through the model variables, applying offsets in the process, for each pass of the vector mapping function.

3.3.3 Interactive Model Insights and Subsequent Approach

The WSPW model built within Vensim demonstrates the potential for an interactive, what-if exploration tool as proposed in Section 3.1.2. The SyntheSim functionality of Vensim enables real-time interactivity with the model, and provides a framework for structuring sliders to modify a base set of data. For manageable data sets, this tool allows for simple, insightful, object-oriented modeling and automatic slider generation through SyntheSim.

Nevertheless, we demonstrate that the interactive capabilities of Vensim are critically limited with large data sets. As these simulations run from a pre-compiled data file, it is necessary to create a data set of sufficient size to allow for scenario exploration. As a result, we decided to pursue a different structure for the What-If Visualization Tool, as explained in Section 3.4. Rather than working from a large, preprocessed data set, we sought to develop a tool that offers the interactive experience of Vensim, yet simplifies the connectivity problem with the data warehouse. The result of these efforts is an Excel-based tool which selects only that data necessary for the particular scenario under consideration. This in effect leads to increased performance and real-time interactive response.

3.4 WVT ARCHITECTURE: EXCEL-BASED MODEL

As described in prior sections, the vision for the What-If Visualization Tool is to enable users to interactively explore the impact of various scenarios on WSPW requirements in a graphical medium. Such a tool would need to embody the following capabilities: (1) graphical rendering of critical parameters and WSPW requirements; (2) user interface display for selection and manipulation of parameters; (3) functionality for calculating WSPW

requirements; (4) database query ability for extracting data from the LRP Database; and (5) the glue to support the interoperability among the diverse components. The proposed architecture for supporting these objectives is exhibited in Figure 20.

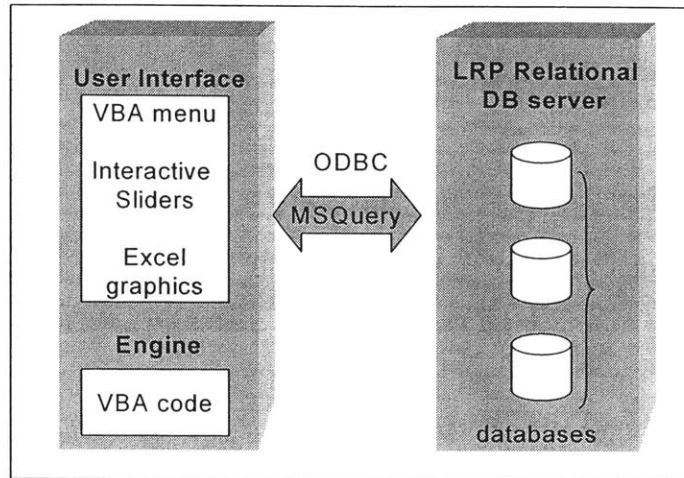


Figure 20: WVT Architecture

The WVT leverages the various capabilities of Excel and its extended functionalities, particularly Visual Basic for Applications (VBA) and Microsoft® (MS) Query. VBA provides the automation “engine” for the tool and enables appealing event-driven user interfaces, including menu displays and interactive sliders. Excel offers both excellent graphical displays and efficient spreadsheet calculations. Finally, MS Query through an ODBC¹ connection provides the means for querying the database and depositing the results within Excel.

The WVT tool, as explained above, has been developed within Excel. This is done essentially for two reasons. First, Excel, as illustrated above, provides requisite functionality for every element of the WVT. Second, Excel is widely used within the SCP team as the de facto tool for spreadsheet computation and graphical display. Hence, intended users could leverage their existing knowledge of Excel and VBA for using and revising the model as necessary. Moreover, the techniques applied in developing the tool, namely the use of VBA

¹ ODBC: Stands for Open DataBase Connectivity, a standard for integrating various database systems. and a spreadsheet tool such as Excel.

automation and MS Query data extraction, could be applied to a host of practical applications requiring interoperability between a relational database and an external application.

3.4.1 Illustration of WVT Interoperability

Perhaps a simple example, adopted from actual WVT code, would best illustrate the interoperability described above. Let us assume that we wish to extract the die size for a given product on a given manufacturing process for the first seven quarters of the LRP horizon. This data is stored in SQL database tables, though we wish to have this data imported into a range of cells in Excel. The intent is to subsequently use this data for calculation of WSPW requirements. The following sections describe the requisite syntax from the perspective of SQL data manipulation language, MS Query object construction, and VBA macros.

SQL Data Manipulation Language

A relational database is so named because the data contained within the database is grouped into several tables that refer to or “relate” to one another. The key idea for using relational databases is to avoid redundancy [14]. Each row in a table represents one of the entities being tracked – in this instance, a single product/process combination. Each column of the table represents the individual pieces of information being tracked for each entity. For our example, the pieces could represent the die size for each quarter in the horizon. In relational database parlance, the rows and columns are referred to as records and fields respectively. Table 4 represents such a simple table.

Prod_ID	Prod_Name	Proc_Name	Q1	Q2	Q3	Q4	Q5	Q6	Q7
1	Alpha	ProcA	200	200	195	195	195	195	195
2	Alpha	ProcB	100	100	100	100	100	100	100
3	Beta	ProcA	350	350	350	350	200	200	200
4	Delta	ProcC	80	80	80	80	80	80	80

Table 4: Sample Database Table: Product_Die_Size

Retrieving data from database tables is accomplished via the Data Manipulation Language (DML) syntax. The reader is referred to Kline et al. for a thorough review of DML syntax [13]. In order to extract the die sizes for product ‘Alpha’ on process ‘ProcB’, the following statements can be utilized:

```
SELECT Q1, Q2, Q3, Q4, Q5, Q6, Q7
FROM Product_Die_Size
WHERE Prod_Name = “Alpha” AND Proc_Name = “ProcB”
```

which yields our desired result:

Q1	Q2	Q3	Q4	Q5	Q6	Q7
100	100	100	100	100	100	100

Table 5: Query Output for WVT Interoperability Example

Microsoft ® Query

MS Query is a freestanding program embedded within Excel that enables querying of a database for subsequent processing within Excel. Internally, Excel creates a *QueryTable* object for every imported table. The function of the *QueryTable* object is to store all information needed for data importation [14]. The key properties of this object are the following:

- *Connection*: contains the path information and password for the database file.
- *CommandText*: contains the actual SQL query.
- *Destination*: denotes the data insertion range within Excel.

Though such an object can be specified manually within Excel using the “Data->Get External Data” pull-down feature, the objects can also be coded directly into a VBA macro, as is done for the WVT.

VBA

The author uses embedded VBA code to generate the entire structure of the WVT. The technique of particular interest to this example is the encapsulation of database querying

capability within the VBA code. This methodology allows for user selection of particular products (components) upon which to perform what-if manipulation. The key advantage of this approach is that it allows for generic code that can be applied repetitively to different user selection criteria.

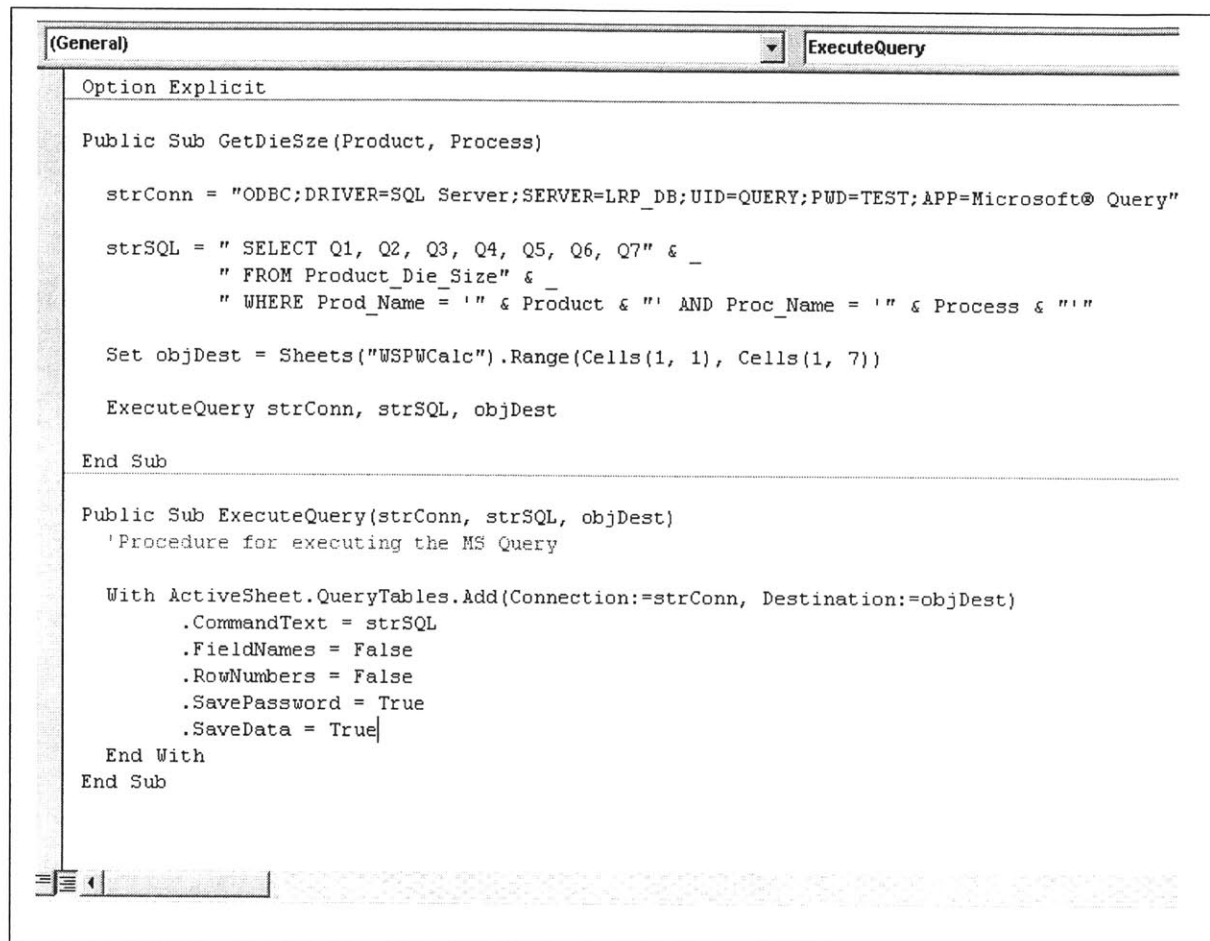


Figure 21: Sample VBA Code for WVT Interoperability Example

Figure 21 shows a screenshot of the VBA code used in the example. Pressing the Alt-F11 key in any application that supports VBA loads the editor as shown in the Figure. Though the structure of the VBA language is beyond the scope of this thesis, Figure 21 nonetheless illustrates the coding of MS Query objects as explained below.

The *Connection*, *CommandText*, and *Destination* properties of the *QueryTable* object are represented by the strings *strConn* and *strSQL*, and the object *objDest* respectively. The string *strConn* contains specifications for the database type, location, and access passwords. The string *strSQL* represents our SQL query, only using the variables “Product” and

“Process” in lieu of the actual names. The object *objDest* denotes the range of cells within Excel where the results of the query (i.e. Table 5 in this example) will be stored. Finally, these values are passed to another generic subroutine (*ExecuteQuery*) where the MS Query object is constructed.

Summary

The aforementioned example demonstrates some of the basic concepts employed within the WVT. Certainly, myriad coding techniques are used in the construction of the various components of the WVT, including the user interface, dynamic components (e.g. sliders), and spreadsheet-on-demand calculations. Nevertheless, the use of dynamic MS Query objects embedded within VBA code has proven to be the key enabler of the WVT, and illustrates a framework for seamlessly importing data from relational databases into Microsoft ® Office applications.

3.5 WVT STRUCTURE AND FUNCTIONALITY

As described in Section 3.4, the What-If Visualization Tool has been constructed in an Excel environment, incorporating Visual Basic for Applications (VBA) for the automation and interaction, Excel for the spreadsheet calculations and graphical rendering of parameters, and MS Query for the live database linking. This section illustrates these features with the exception of MS Query, which is treated in Section 3.4. Hence, the goal of the following discussion is to give the reader a visual feel for the structure and functionality of the tool.

3.5.1 WVT User Interface

The user interface for the WVT is displayed in Figure 22. This interface, the “MainMenu”, serves as the interactive environment wherein the user selects the component for the subsequent what-if analysis. A unique component is identified by the particular LRP database, fab process, and dot process. Recall from Section 3.1.1 that with the current what-if process, the user is required to create a new database. With the WVT, the user need only select the base database for scenario experimentation.

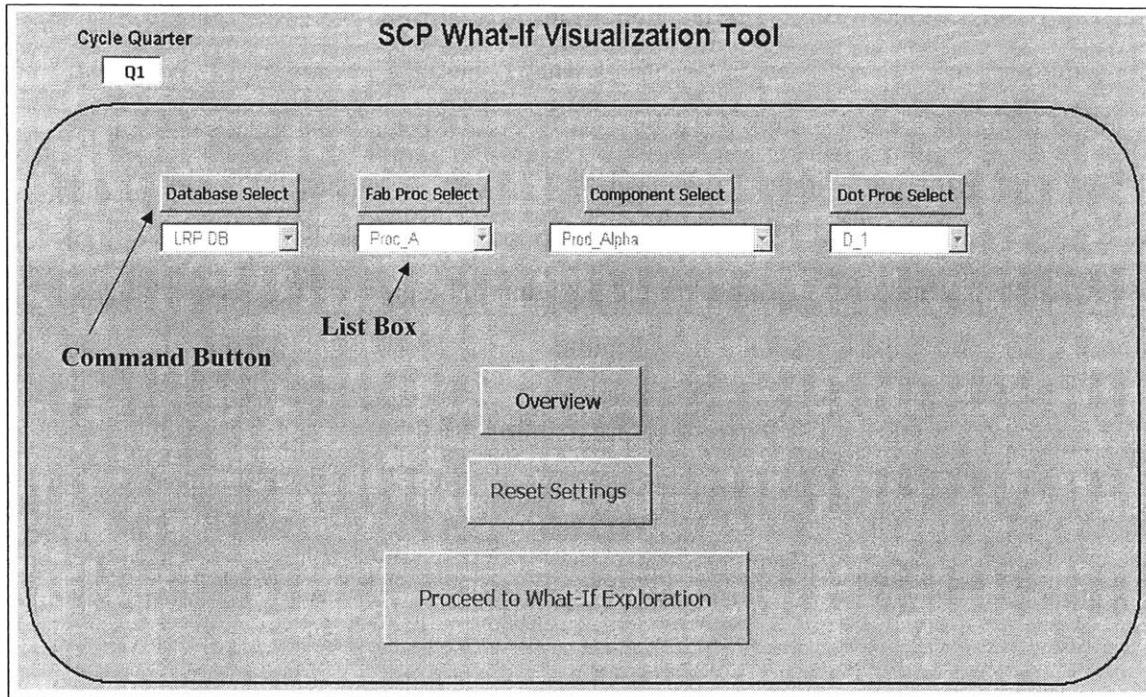


Figure 22: WVT User Interface

The user specifies the parameters for the scenario evaluation by sequentially populating the database, fab process, component, and dot process fields in that order. The user accomplishes this by first pressing the appropriate command button¹ as shown in the Figure. The command button then executes an event procedure to extract the relevant information (using MS Query and VBA code) from the LRP database, displaying the result in the collapsible list box² beneath the particular button. Upon making a selection from the list box, the field is populated and the subsequent button is enabled for selection. This disabling/enabling property of the input devices (i.e., as denoted by “grayed out” or “bold” labeling respectively) serves to constrain the user from making selections out of sequence.

The “Cycle Quarter” field in the upper left corner of the interface displays the quarter within which to apply the beginning-on-hand initial conditions for the strategic inventory points. This is critical from the standpoint of WSPW calculations, as explained in Section 3.2.2.

¹ Command button: a VBA control object that facilitates the programming of event procedures.

² List box: refers to a VBA object that allows a user to choose among items in a pull-down list.

The WVT runs through some simple calculations to determine the cycle quarter, prompting the user to select between alternatives when necessary as shown in Figure 23. Finally, once all fields are populated, the user launches the What-If Visualization Tool calculations by pressing the “Proceed to What-If Exploration” command button. Figure 24 shows a representative interface where all fields are populated. Aside from these essential process steps, the user can also invoke a simple help dialog by pressing the “Overview” button or reset the conditions with the “Reset” button.

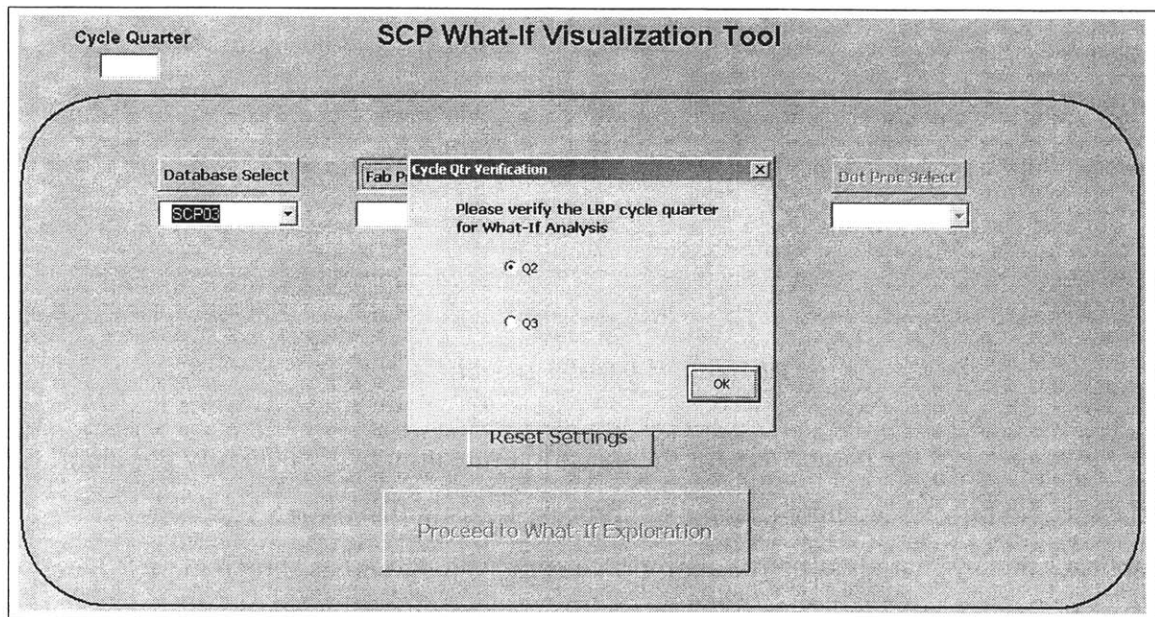


Figure 23: WVT User Interface - Cycle Quarter Selection Dialog

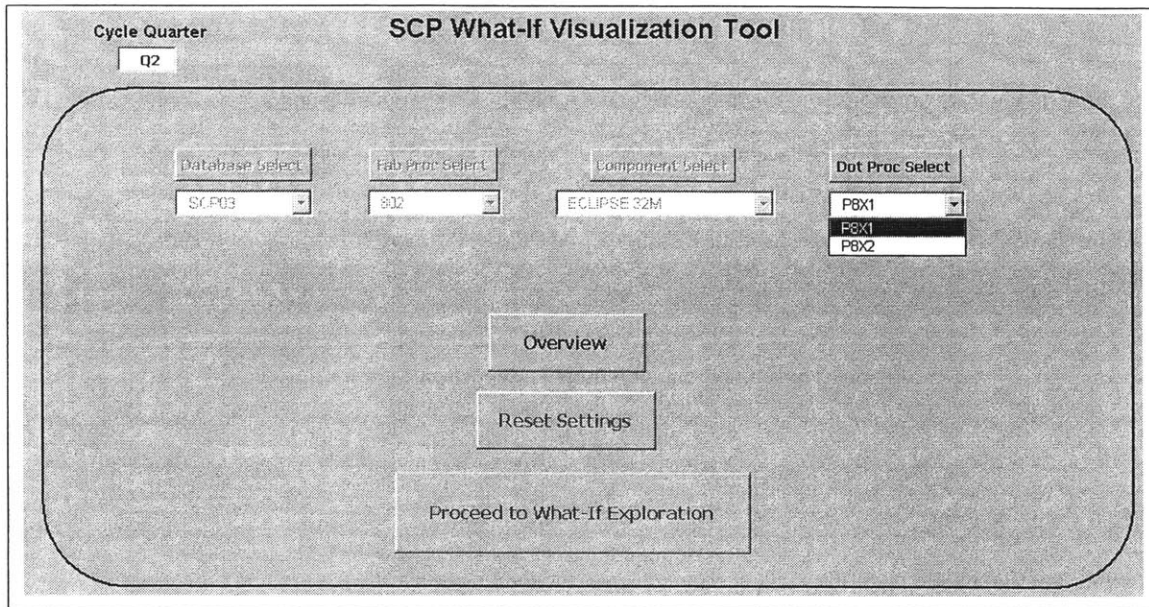


Figure 24: WVT User Interface - Field Population

3.5.2 Spreadsheet Calculations

The parameters specified in the “MainMenu” as described in the prior section provide sufficient detail for WSPW calculations. These parameters are subsequently used in structured SQL queries to obtain WSPW parameters from the LRP database. Indeed, part of this research effort has involved identifying the appropriate database tables and crafting queries to extract the desired information.

The WSPW calculations are executed in spreadsheets as shown in Figure 25 below, where the rows represent particular variables or parameters and the columns represent the quarters of the LRP horizon. As each what-if scenario is unique, spreadsheet calculations such as those exhibited in Figure 25 need to be built up with each scenario run (i.e. unique menu combination). VBA code is employed to automate this process. Note that while some parameters such as fab yield (FY) are extracted from the LRP database, variables such as die per wafer (DPW) are calculated using Excel formulas. Figure 26 offers sample VBA code to illustrate the key logical differences in the syntax between these two situations. As gleaned from the code, the FY parameter involves creating an MS Query object with an Excel range destination, while the DPW variable populates a range of cells with a formula.

Parameter Time	Q1'02	Q2'02	Q3'02	Q4'02	Q1'03	Q2'03	Q3'03	Q4'03
Dmd	8000	12000	18000	19000	17000	16000	14000	7000
Dmd_What-If	8200	12400	18300	19000	17500	16000	14000	7000
Dmd_Delta	200	400	300	0	500	0	0	0
DC_Build	8000	14000	18000	16000	14000	12000	8000	0
Die Size	100	100	100	100	100	100	100	100
Throughput Time	1.5	1.4	1.4	1.2	1.2	1.2	1.2	1.2
Die_Bank_Bld	15000	16000	14000	12000	11000	8000	4000	0
DPW	200	200	250	250	250	250	250	250
FY	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
WSPW	75	80	56	48	44	32	16	0

Figure 25: Sample Spreadsheet Calculation Segment

```

Public Sub GetFY()

    Sheets("FSMCalc").Select
    Cells(19, 1) = "FY"

    strConn = "ODBC;DRIVER=SQL Server;SERVER=LRP_DB;APP=Microsoft® Query"
    strSQL = "USE " & Db & _
        " SELECT q1, q2, q3, q4, q5, q6, q7, q8" & _
        " FROM fab_yield WHERE fab = '" & Fab & "'" AND fab_proc = '" & FabProc & "'"

    Set objDest = Sheets("FSMCalc").Range(Cells(19, 2), Cells(19, 9))

    ExecuteQuery strConn, strSQL, objDest

End Sub

Public Sub CalcDPW()

    Application.ScreenUpdating = False
    Dim i As Integer
    Sheets("FSMCalc").Select
    Cells(20, 1) = "DPW"

    For i = 2 To 9
        Cells(20, i).FormulaR1C1 = "=DPW(R[-2]C, R[-6]C, R[-9]C)"
    Next i

End Sub

```

Figure 26: Sample VBA Code for Parameters and Variable Population

Finally, it is worth noting that the spreadsheet calculations are complicated by the change in product form function from packaged components to die as we advance up the supply chain. In keeping with the level of aggregation desired by the SCP analysts, users select the product “component” as opposed to the product and package. It is then the component demand that is modified in subsequent scenario exploration. This approach is supported by

the precept that it is important to provide data whose “granularity” closely matches that currently in use. However, the model must perform inventory calculations, as explained in Section 3.2.2, using the packaged product. The net effect is that the WVT includes the ability to dynamically create calculation ranges in Excel for each component/package combination.

3.5.3 Dynamic Scenario Exploration

The dynamic exploration sheet lends the manipulative power to the What-If Visualization Tool. The aforementioned spreadsheets, menu interface, VBA code, and MS Query objects represent the enabling technologies. Indeed, once the user selects the desired semiconductor component parameters from the “MainMenu”, that user is directed to the “WhatIf” sheet, a portion of which is depicted in Figure 27.

The “WhatIf” sheet contains embedded line graphs as well as two key VBA controls – vertical sliders and horizontal scrollbars as revealed in Figure 27. Like the “MainMenu” sheet, this worksheet is not reconstructed with each scenario configuration. The graphs are linked to ranges within the supporting Excel spreadsheets, and depict the values for the base case, the what-if case, and the delta. This simple convention is consistent with existing scenario analysis practices.

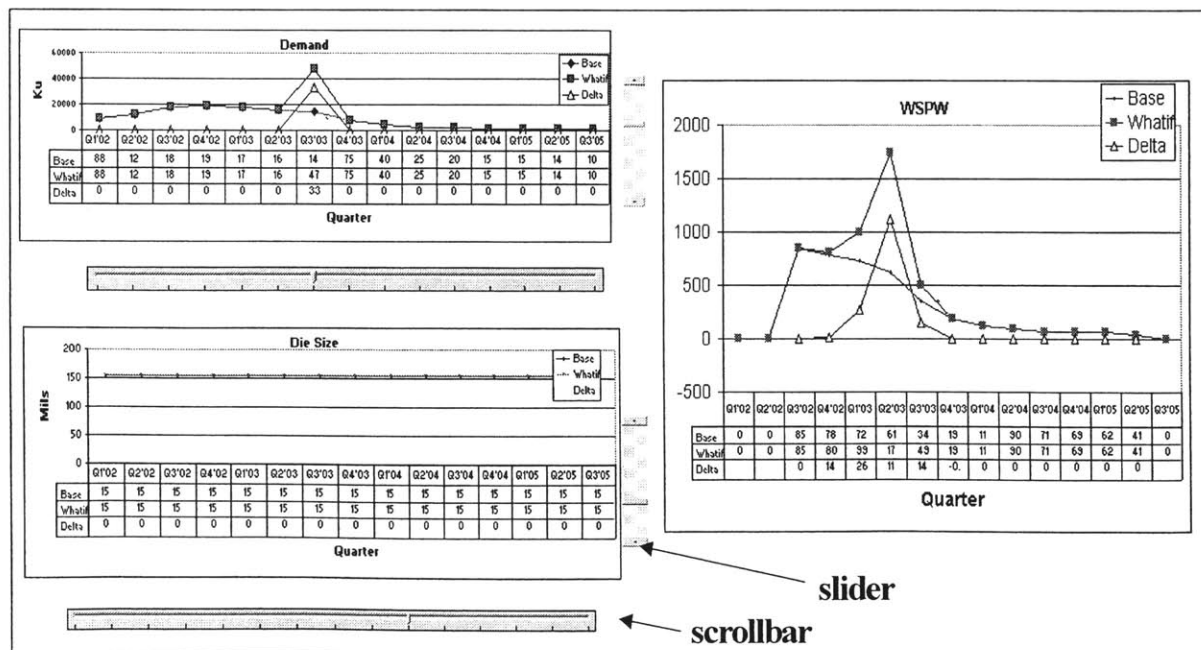


Figure 27: WVT Dynamic Exploration

The slider and scrollbar allow the user to explore the base case. First, the scrollbar is positioned or “dragged and released” to a particular quarter in the LRP horizon. Next, the dynamic slider is adjusted manually by the user. The slider is dynamic in that every slight movement results in a perceptually immediate change in the parameter being manipulated. The three parameters available to the user are the demand, the die size, and the die yield (not shown). Moreover, the effects of the parameter change on WSPW are also interpreted. As shown in the figure, an increase in the demand for Q3’03 has led to an increase in WSPW from Q1’03 through Q3’03. This is what we would expect given the sensitivity of WSPW to throughput times as explained in Section 3.2.2.

Finally, the underlying logic of the slider and scrollbar controls deserves mention. These tools are linked to a fixed range of cells within the WSPW calculation sheets. The position of the scrollbar identifies the particular cell within the range, while the adjustment of the slider determines the scalar value within that cell. Consequently, the scenario parameter (demand, die size, or die yield) is the product of the base case value and the scalar, which is initially set to a value of one.

3.6 CHAPTER SUMMARY

The What-If Visualization Tool is inspired by the empirical observation that the existing mechanics for conducting scenario analyses is somewhat process-intensive. The case for a streamlined scenario exploration process, centered on interactive manipulation of scenario variables, is presented. The general semiconductor supply chain is explained for the purpose of demonstrating the type of calculations underpinning the WSPW model. The initial approach for constructing the what-if tool, using the Vensim simulation environment, is illustrated as it serves to hone our vision for the functionality of such a tool – namely as a client processor for WSPW calculations, as opposed to a data warehouse. Finally, the WVT is presented, focusing on its architecture, structure, and usability components.

4 DATA VISUALIZATION WITH TREEMAP 3.0

The advantages of using treemaps for visualization of complex, hierarchical data structures are readily observable. Chapter Two illustrates some of these concepts, and elucidates the general layout algorithm that is central to the space-filling approach used to construct the densely packed yet resolvable data elements. For this thesis work, Treemap 3.0, a Java application developed by the University of Maryland's Human Computer Interaction Lab, has been utilized [32]. This chapter briefly presents the Treemap 3.0 tool in the context of the human interaction model, and discusses the methodology pursued in this work to improve its utility for the strategic capacity planning process.

4.1 HUMAN INTERACTION MODEL – TREEMAP 3.0

Treemap 3.0 contains a simple, efficient, menu-driven user interface as shown in Figure 28.

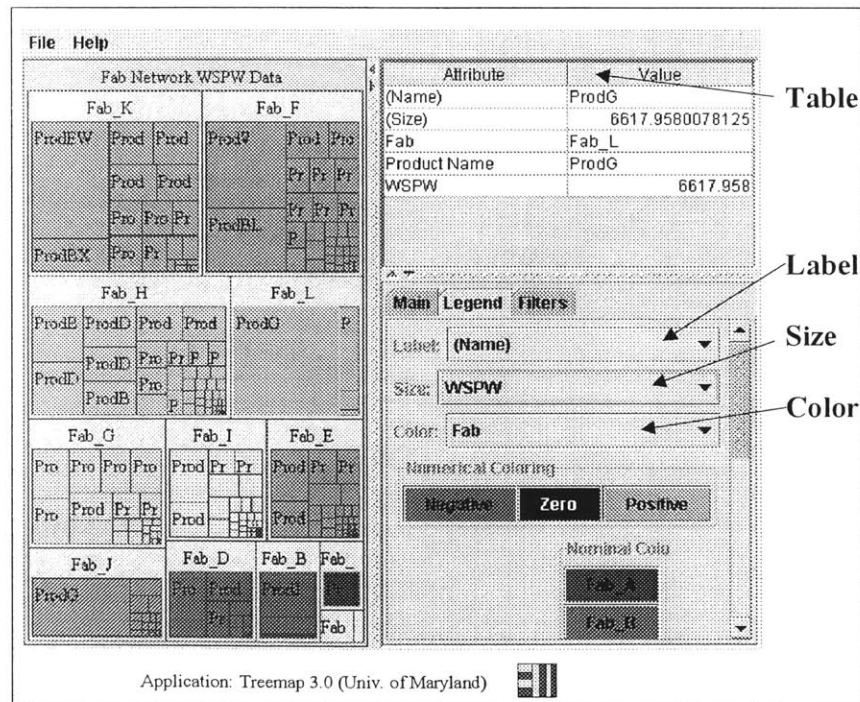
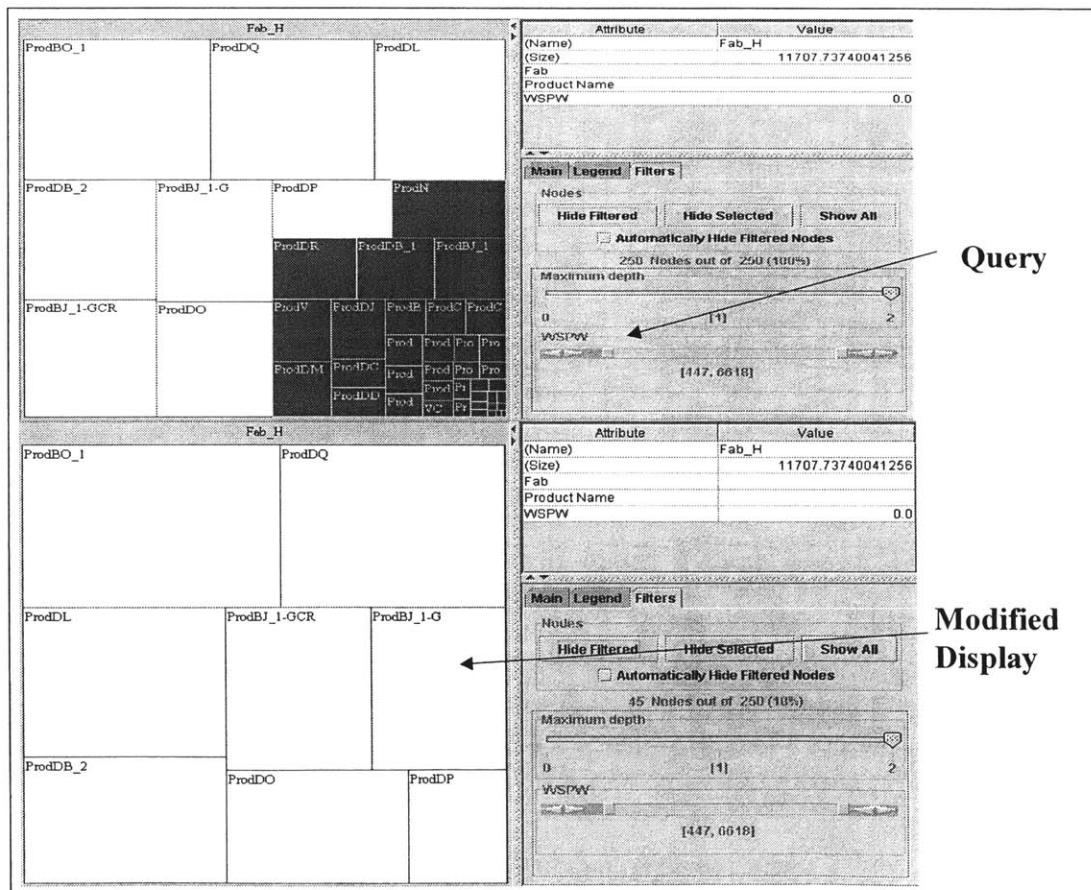


Figure 28: Treemap 3.0 User Interface

The three main areas of the screen are the treemap display, the data table, and the interface menu. The treemap display represents the detailed, hierarchical data as specified in the interface menu with the parameters label, size, and color. For instance, the user simply selects a variable from the “size” menu for which the user would like to see a treemap drawn. In this example, the number of WSPW required for the respective component sizes the rectangles. Once the user selects a particular rectangular element in the display, the data table depicts all the available information stored in the data set.



Users gain a real sense of control with dynamic queries. As Shneiderman asserts, users can “quickly perceive patterns in data, ‘fly through’ data by adjusting sliders, and rapidly generate new queries based on what they discover through incidental learning” [6].

Finally, users can “double-click” on an individual rectangle or its parent to zoom into the data, as shown in Figure 30 below. The elements (leaves) within the particular node (branch) are automatically resized to fit the display area. This feature is particularly useful for drilling down on display elements that may be minimized due to outliers, enabling the user to explore the data.

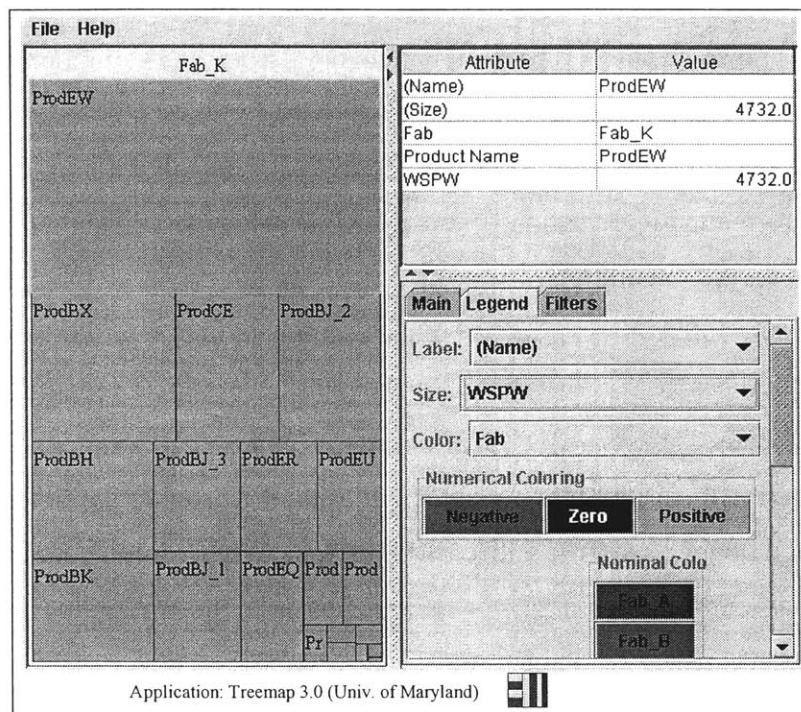


Figure 30: Demonstration of Zoom Feature of Treemap 3.0 (base display Fig. 28)

Invoking the human interaction framework, we note that the sequence of transforming data into information and insight proceeds from *raw data* to *data tables* to *view structures* and finally to *views* as depicted in Figure 5. As the aforementioned features exemplify, this treemap tool is equipped to support the *view structures* to *views* link effectively, thus facilitating the act of “thinking about the data.” However, this tool works from a data set (in Excel) created by the user. Hence, a methodology is needed to integrate the other stages.

4.2 THE TREEMAP DATA MACRO

Information visualization is about using tools that facilitate rapid, accurate insight into data. As stated in the opening chapter, the author posits that treemap displays could augment the strategic capacity decision making process by presenting information in a compelling way.

Treemaps are applicable in the SCP context as the long range planning information is by nature detailed and hierarchical. The atomic unit of display is characteristically the end product (component), of which there are hundreds. These components can be aggregated by fabrication process, factory, or division depending on the problem being addressed through the display. Finally the individual components need to be sized, labeled, and colored. For example, components can be sized by WSPW, unit demand, or average selling price to name a few. The challenge is to make this data available to the user.

The specific approach pursued by this work has been to develop an automated data-gathering tool to pull formatted information from the LRP Database for display in Treemap 3.0. Such a tool supports the integration of *data tables* to *view structures* in the context of the human interaction model. Interestingly, the techniques employed in this tool mirror those of the WVT. A VBA macro has been coded in an Excel environment to fetch the information from tables in the LRP database using MS Query objects. Upon executing the macro, a spreadsheet is populated with various component records and their respective fields, such as WSPW for a designated quarter, over the entire LRP horizon. When running Treemap 3.0, the user can select a particular sizing parameter for any quarter in the horizon. Thus, the information visualization experience is much improved.

4.3 TREEMAP DISPLAY OF LRP DATA

Aggregate long-range plan information is both hierarchical and detailed. As discussed in prior sections, the data stored in the data warehouse comprises product-level demand and associated WSPW transformation parameters, as well as attribute data such as product division and factory. Though displaying this data in a single view is problematic for traditional display techniques, treemaps are ideally suited for such a task.

Example

One of the most oft-used applications of treemaps within SCP has been the organization of products by fab or product division, with the sizing parameter denoted as WSPW, unit demand, or revenue. Figure 31 represents one such example, using WSPW as the spatial variable. Note the relative ordering of both nodes (fabs) and leaves (products) within the display. There is a natural placement of larger shapes in the upper left corner, with smaller shapes in the lower right corner. As discussed in Chapter 2, this affords rapid detection of the key drivers for WSPW in this instance. Based on a size comparison of Figure 31, it appears as though FabK and FabG have similar WSPW requirements, with FabE slightly larger.

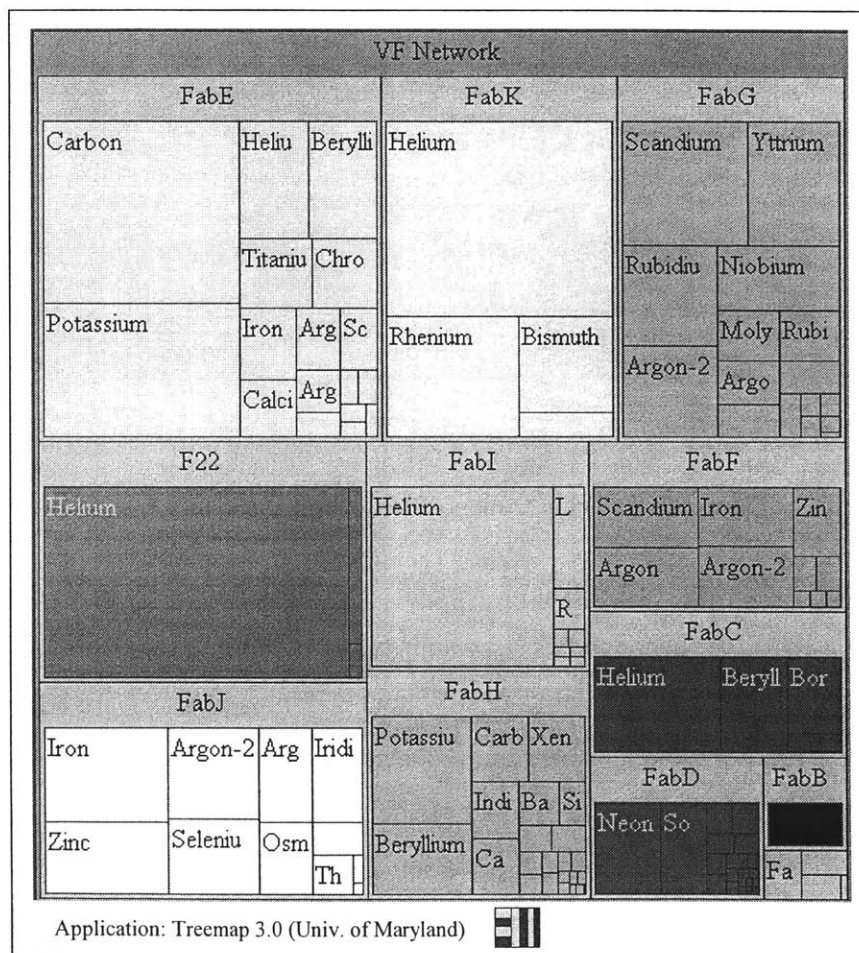


Figure 31: Treemap Display - Size by WSPW

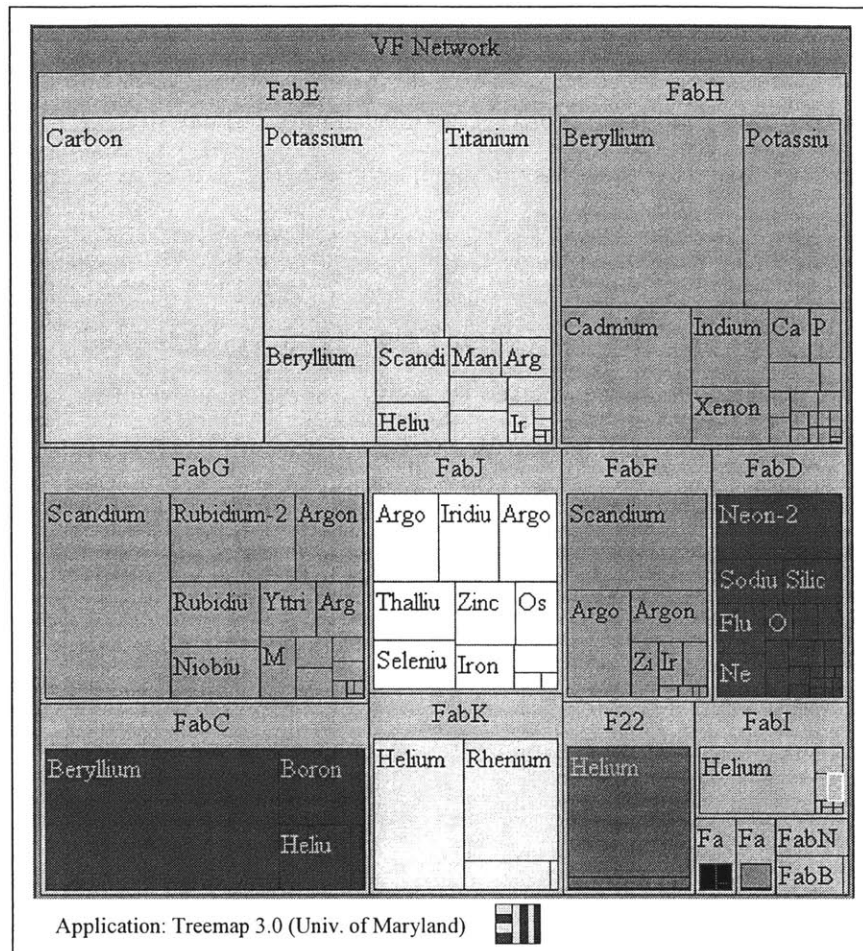


Figure 32: Treemap Display - Size by Unit Demand

In Figure 32, the sizing parameter is unit demand. FabE has increased in size relative to the other fabs, while FabK for instance has been supplanted by FabH. One conclusion to be drawn from these observations is that FabK WSPW conversion parameters are less favorable than those of FabH. In other words, FabK requires relatively more wafer starts to support a given demand. This could be due to the product mix (e.g. larger die sizes require more WSPW), the wafer size mix, or the fab yield for example.

The preceding example illustrates the speed with which one can gain insights from treemap displays, as well as the type of discussions that such displays could stimulate. Given the time constraints emblematic of decision-making forums, and the wealth of information available, these displays have proven to be a powerful addition to the arsenal of visual display tools.

5 RESULTS AND ANALYSIS

The previous chapters lay the foundation for an understanding of the underlying motivation, logic, and structure of the What-If Visualization and Treemap display tools. This chapter considers these tools in a practical environment, offering insight into their utility and range of applicability. Hence, the WVT is analyzed in the context of numerical accuracy and conformance to effective user interface design. Treemaps are then empirically evaluated for their ability to visualize information relevant to the strategic capacity planning problem.

5.1 WHAT-IF VISUALIZATION TOOL

5.1.1 WVT Data Validation

Model Assumptions

The mathematical structure of the WVT has been designed to emulate the calculations of the LRP database, as disclosed in Chapter 3. Indeed, the WVT in a sense serves to offload the WSPW calculations of the LRP database, reducing its function to that of a data repository. Nevertheless, a notable simplification of the LRP model has been undertaken to facilitate a more manageable spreadsheet solution to the WSPW calculations. This simplification is the use of “virtual factory”¹ parameters for fab yield (FY), die yield (DY), and throughput time (TPT).

The WSPW capacity required to meet a given component demand is dependent upon the component’s specific attributes (e.g. die size, inventory strategies) as well as the factors of production (e.g. fab yield). As these factors of production vary by production site and process, the routing strategy chosen for a particular product affects the requisite WSPW capacity for a given demand. Under certain conditions, however, the specificity of such factory parameters is not critical to the decision at hand. In other words, the expected error of using one prototypical set of factory parameters for the WSPW calculation is insignificant. It is for such decisions that so-called virtual factory (VF) parameters can be employed to

¹ Virtual factory: refers to the factory network as a single, “virtual” factory.

simplify the calculations. VF parameters treat the factory network as a single closed system with one set of factory parameters.

Data Analysis

Data validation for the WVT involves a paired comparison of WSPW calculations between the WVT and the LRP Database. An initial set of ten components has been randomly selected, ensuring that the range of wafer diameters (200mm and 300mm) and throughput times (TPT <1, TPT >1) are represented. This data is reflected in Table 6.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	
ProdA	0	0	0	55	59	95	115	68	6	WVT
	0	0	3	59	61	97	112	76	7	LRP
	NA	NA	NA	-5.99%	-2.94%	-1.60%	2.49%	-11.08%	-14.61%	% error
ProdB	0	41	104	436	434	482	424	187	16	WVT
	6	40	123	460	436	476	396	209	18	LRP
	NA	1.60%	-15.40%	-5.22%	-0.46%	1.18%	7.00%	-10.33%	-13.50%	% error
ProdB	603	247	43	0	0	0	0	0	0	WVT
	610	250	43	0	0	0	0	0	0	LRP
	-1.18%	-1.17%	-0.07%	NA	NA	NA	NA	NA	NA	% error
ProdB	578	339	235	197	184	105	20	0	0	WVT
	574	355	226	195	182	104	20	0	0	LRP
	0.71%	-4.51%	3.85%	0.91%	1.32%	1.00%	-0.39%	NA	NA	% error
ProdB	337	280	286	270	171	78	9	0	0	WVT
	344	282	290	267	175	81	11	0	0	LRP
	-2.17%	-0.75%	-1.47%	1.04%	-2.26%	-3.82%	-21.04%	NA	NA	% error
ProdB	0	0	0	25	2232	4273	4909	4328	2713	WVT
	0	0	0	25	2268	4284	4919	4340	2737	LRP
	NA	NA	NA	-0.17%	-1.59%	-0.25%	-0.21%	-0.27%	-0.87%	% error
ProdB	235	25	0	0	0	0	0	0	0	WVT
	303	33	0	0	0	0	0	0	0	LRP
	-22.44%	-23.74%	NA	NA	NA	NA	NA	NA	NA	% error
ProdB	4484	2484	3060	2090	275	0	0	0	0	WVT
	5042	2791	3428	2340	299	0	0	0	0	LRP
	-11.07%	-11.01%	-10.74%	-10.70%	-8.18%	NA	NA	NA	NA	% error
ProdB	849	726	477	141	14	0	0	0	0	WVT
	949	813	533	157	16	0	0	0	0	LRP
	-10.56%	-10.67%	-10.58%	-10.39%	-10.30%	NA	NA	NA	NA	% error
ProdB	172	174	159	140	134	119	100	83	77	WVT
	190	190	174	156	146	131	110	90	85	LRP
	-9.58%	-8.38%	-8.34%	-10.14%	-8.43%	-9.42%	-9.18%	-7.61%	-9.19%	% error

Table 6: Paired Comparison of WVT/LRP WSPW Calculations

The data above reveal an average discrepancy in WSPW calculations of a negative 6%, with a standard deviation of nearly 7%. Certainly, some variance is expected given the

variable truncation known to exist within the LRP database as well as the use of VF values in the WVT. However, the WVT consistently calculates lower WSPW requirements when compared to the database, and in some cases (e.g., ProdG) is off by some twenty percent. A priori, the WSPW under-reporting problem is believed to be due to data truncation and component requirements (e.g., engineering wafer demand) not reflected in the base equations, while the large percentage deviations are assumed the result of interactions between the build plan horizon (0 through 9 months) and the LRP horizon (9 months through 5 years). While the errors may be acceptable for scenario analysis anchored on high demand drivers, the author recommends that the root cause of the data variance be pursued.

The data discrepancies cited above notwithstanding, the What-If Visualization Tool provides a useful setting for a certain class of components. The 80/20 rule, wherein 20% of the components are responsible for 80% of the capacity requirements, is certainly relevant for semiconductor manufacturing. In general, it is for these 20% that the WVT could provide augmented exploration capacity for the SCP analyst.

5.1.2 WVT Usability Characteristics

The What-If Visualization Tool applies the usability constructs of metaphors, S-R compatibility, affordances, and constraints as presented in Section 2.4. The following discussion highlights some aspects of their use in the WVT.

Metaphors

The main menu of the WVT (ref. Figure 22) has been designed with emphasis on simplicity and ease of learning. The command buttons and drop-down menu lists are objects familiar to the computer literate, inviting the user to test the functionality of these interactive objects. The scenario exploration screen contains temporal line graphs, perhaps one of the most common graphing methods in use. Finally, the slider object is familiar to those accustomed to using any type of office productivity software.

S-R Compatibility

Stimulus and response (S-R) compatibility refers to the effectiveness of a display in conveying the connection between spatially separate objects. In the WVT main menu, command buttons are paired with their respective menu lists. Furthermore, these pairs are

arranged from left to right in an ordered sequence, providing a natural process flow from left to right, and top to bottom. The user initiates the pull-down menu by activating the button. Hence, the button populates the pull-down list by querying the database. Once the user makes a selection from this list, the next (to the right) button becomes activated, allowing the population of the next pull-down menu.

The scenario exploration screen perhaps embodies the most effect use of S-R compatibility. The scroll bars have been arranged such that the respective tick marks are aligned with the horizontal axis markers on the associated line graph, as shown in Figure 33.

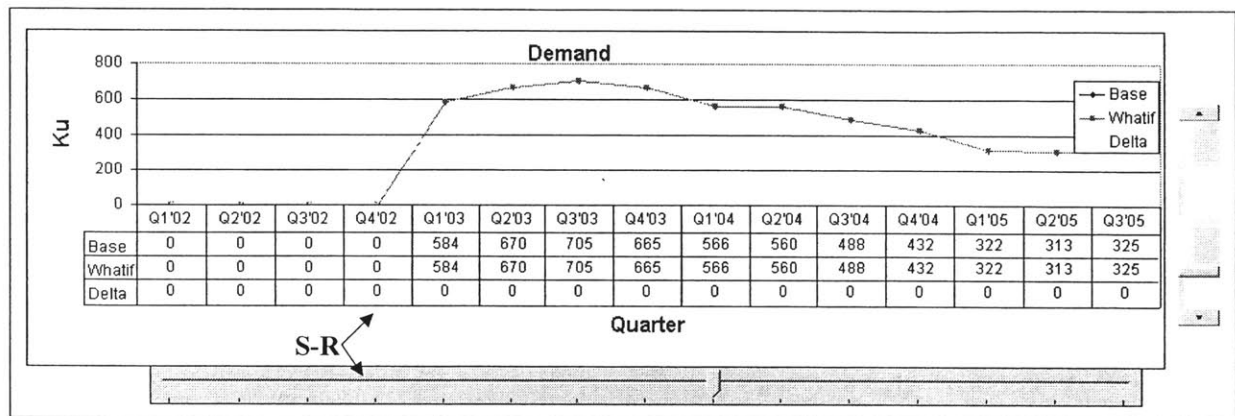


Figure 33: Demonstration of S-R Compatibility - WVT Example

Affordances and Constraints

Affordances encourage the user to perform an action, while constraints “limit” the possibilities for action. As discussed above, the main menu screen contains a number of command buttons and menu lists. These icons invite the user to activate the objects through mouse clicking or other means. The menu lists also afford item selection by pre-loading the selection list in lieu of manual data entry by the user. Finally, the scroll bars and sliders encourage direct manipulation.

Constraints signal to the user the available choices for action in addition to protecting the system from improper sequences. The enabling/disabling feature of the menu screen interface components, wherein the object labels are highlighted in bold face or gray type respectively, illustrates this concept. Indeed, this arrangement ensures that queries are not executed prior to specifying all search parameters.

5.1.3 Relationship Detection with the WVT

The myriad benefits of an interactive, graphic, intuitive visualization tool are enumerated in Chapter 2. The third chapter discusses the potential benefits to be derived from the What-If Visualization Tool, including a reduction in time for scenario analysis, an alternative means for computing WSPW requirements, and a medium within which to foster an understanding of the dynamics of WSPW calculations. This last benefit is illustrated in Figure 34.

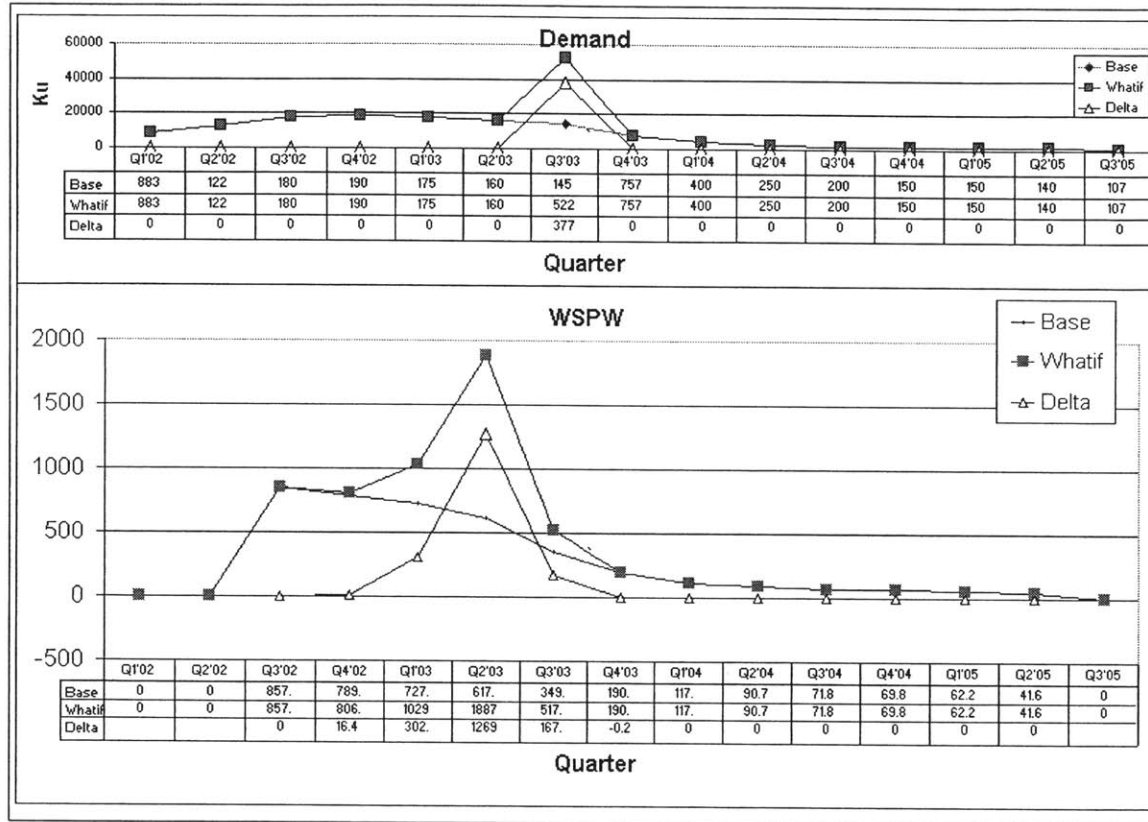


Figure 34: WVT - Nonlinear Relationship Depiction

Figure 34 exhibits a scenario where demand for Q3'03 is increased from 145 to 522 (260%). This leads to an increase in WSPW distributed over four quarters, from Q4'02 through Q3'03. As discussed in Section 3.2, the throughput time effects are at work here. In this particular example, the greatest effect on WSPW is in the quarter prior to demand – Q2'03. In scenarios where peak ramp is being evaluated, such what-if experimentation could be of real value. Indeed, adding a line to the graph representing peak capacity (according to current plan) could facilitate dynamic, user-controlled sensitivity analysis using the interactive sliders.

5.2 TREEMAP DISPLAYS

The treemap displays have been used to generate visual displays of detailed, hierarchical data as described in Section 4.3. These displays have proven effective in communicating information to key decision makers involved in complex decisions, and have stimulated thought on future applications for treemaps within the organization.

Figure 35 represents the prototypical treemap application, wherein a side-by-side comparison of treemaps is presented – one sized by demand, the other by WSPW requirement. Such a display is rich in information, conveying the high-level view of the manufacturing scene. Moreover, the drivers of demand, as well as wafer starts, immediately “jump out” at the viewer, prompting discussion on risk abatement strategies as well as actions to reduce required capacity.

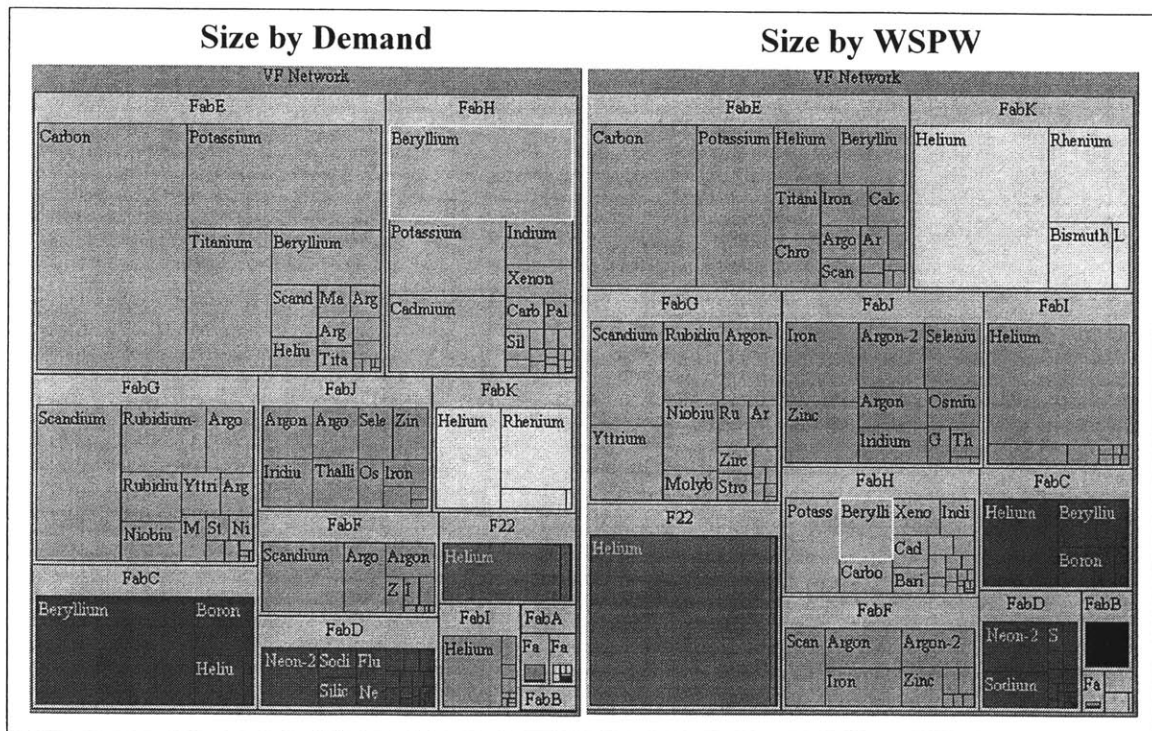


Figure 35: Treemap Paired Comparison

5.2.1 Emergence of a Dominant Design

Though the applicability of treemaps to structured, detailed data is readily apparent, it was through a process of trial and error, brainstorming, and customer feedback that a standard set of display archetypes has evolved. These are represented in Table 7.

Hierarchy	Leaf (component) Size	Leaf (component) Color
Fab	Revenue	Component
Division	WSPW	Mfg Process
	Unit demand	Fab
		%chg in revenue
		%chg in WSPW
		%chg in demand

Table 7: Treemaps in Use - Parameter Mix

Hierarchy

Hierarchy represents the structure, or the “branches” of the treemap. Within each branch or node, the treemap algorithm draws the “leaves” or “children” of the node. In Figure 35, the nodes are specified as fabs, and the children as products. It is not surprising that “Fab” and “Division” emerged as the de facto hierarchy display types, as these two levels of aggregation are elemental to the strategic capacity planning problem.

Leaf Attributes

The leaves represent the basic element of the treemap displays. The squarification algorithm that is the core of treemap displays is amenable to large collections of leaves. Similarly, the component or unit product represents the building block of the capacity plan. Demand forecasts from the various divisions are submitted at the unit demand level, and then aggregated through the long range planning process. As such components typically number in the hundreds for a diverse semiconductor company such as Intel, the treemap depicts the data well. The decision then hinges on the leaf size and color.

The size of the leaves or nodes is intended to support discovery and insight into the data through relative comparison. For instance, product “Carbon” of FabE in the left diagram of Figure 35 has the greatest unit demand for the particular time period in the forecast. The right

side of the Figure displays the components sized by WSPW. Both displays are important to key decisions in the LRP. For instance, fab managers may be more concerned with WSPW requirements, while marketing managers may be more interested in the unit demand mix. One sizing parameter that has garnered interest is revenue by product, defined as the product of unit demand and average selling price for a particular component.

Color can be used to communicate additional information about the data elements (i.e. components), in effect adding a third dimension to the conveyed information. Color can be used both for categorical and continuous variable display. Categorical color depiction involves choosing from a list of distinct colors equal to the number of categories. For instance, treemaps have been constructed using “manufacturing process” as the ordinal variable. The fab manager then would have insight into the apportionment of capacity across the manufacturing processes within one’s fab, in addition to the concentration of capacity among the components to be manufactured. Another effective use of ordinal coloring is for accentuating the hierarchies. Stated differently, one could color the components by the same variable used for the hierarchy, thus enhancing the ability to draw conclusions from the data. Finally, color can be used for continuous variable display. The green/red scale has been used to convey relative increase/decrease of a particular parameter over time. For instance, we could use continuous coloring to represent the percent increase/decrease of demand vis-à-vis the prior forecast plan. The contrast and vividness of the hues would indicate the degree of change. In sum, the color attribute of treemap leaves offer a compelling means to improve the display of information.

5.2.2 Automating Treemap Generation

Prior to the emergence of a dominant design for treemap parameters, the displays had been generated in an ad-hoc, manually intensive fashion. Subsequent to feedback from decision-making forums where treemaps have been introduced, it became clear that the type of information displayed could be categorized as shown in Table 7. Thus, an Excel-based macro has been developed to automate the task of populating a formatted spreadsheet. This tool, much like the WVT, presents the user with a simple menu selection for selecting database parameters. However, it differs from the WVT in that all data needed for user exploration is extracted. The user then uses the treemap selection window to select the

particular parameters for display. The WVT, on the other hand, selectively queries the database as its data needs are much more detailed.

5.3 SUMMARY

This chapter illustrates the features and use of both the What-If Visualization Tool and the Treemap display tool in the strategic capacity planning process. The WVT is an interactive, graphic tool that enables users to dynamically modulate demand, die size, and die yield for a particular component in the LRP Database. The key insight in this instance is the effect on WSPW requirements. The Treemaps, on the other hand, have been designed to extract and convey aggregate information in an efficient, space-optimizing manner. While this tool does allow for dynamic manipulation of the views, it does not facilitate modification of the underlying parameters. Nevertheless, both tools represent distinct methodologies for facilitating the process of discovery, decision-making, and explanation.

This Page Intentionally Left Blank

6 CONCLUSIONS AND RECOMMENDATIONS

6.1 SUMMARY OF FINDINGS

This research work has been inspired by the hypothesis that innovative visualization techniques offer an opportunity for enhanced comprehension of data. This comprehension arises not only from the presentation of data, but from all stages of the data visualization experience, including the collection of data, organization of data, rearrangement of data, and transformation of data into information. The challenge of information visualization techniques is to accomplish this end while providing a robust and simple interface design.

The What-If Visualization Tool has been developed to enable the comprehension of complex relationships in the wafer start calculation model. The current scenario generation process requires the capacity analyst to become immersed in the details of the data, transcribing and aggregating the data into tables. The WVT allows the user to proceed directly to the data manipulation sub phase upon selection of scenario parameters. This encourages the exploration of varied scenarios, and facilitates an understanding of the relationship between key product/process/manufacturing parameters and wafer start requirements.

The treemap display methodology has been utilized to display hierarchical, complex data in a concise and compelling manner. The space-filling approach of treemaps ensures that the entire display space is functional. The squarification algorithms do well to render even the smallest rectangular objects visible. Finally, the method of enclosing objects within objects conveys the organizational structure of the data. This confluence of capabilities gives rise to a unique visualization medium that amplifies cognition. Indeed, treemaps have been shown to communicate key messages of the strategic capacity planning process and to describe and understand dependencies of the many elements in the manufacturing system.

The visualization tools used in this research effort focus on different levels of the human interaction model. The strength of the WVT lies in its connectivity and automation, transforming raw data into visual structures. This technology is supported primarily through the MS Query objects, VBA automation, and the spreadsheet and graphing capabilities of

Excel. The user retains the perception of perpetual control of the system, loading data through a dynamic menu interface and modifying the resulting graphs through interactive sliders. The treemap tool, by contrast, is an offline visual manipulation tool. After loading the data into a file, the user selects the fields for display and modifies the view through filtering and zooming techniques. Displays judged rich in information for decision-making forums are subsequently captured for display in a static medium. Though automation has been developed to speed the creation of data files, the power in treemaps lie in their innovative means of conveying information.

6.2 SUGGESTIONS FOR FURTHER RESEARCH

The integration of disparate software tools into enterprise, internet-enabled systems is improving the potential for visualization tools to gain prominence in the future. Powersim simulation solutions, for example, utilize enterprise data stored in ERP (Enterprise Resource Planning) systems. Such simulation applications can include a customized, web-based interface allowing managers to collaborate online as they explore various scenarios [20]. The migration of the LRP Database to a VB.net platform presents a similar opportunity for interactive tools such as the WVT. With VB.net technology, interactive, dynamic structure could be added to allow users to explore scenarios in a web-enabled environment. Moreover, there are opportunities to create more advanced interactive environments, using simulations or complex mathematical models.

Similarly, innovative display methodologies such as treemaps could be buttressed with dynamic links. One such manifestation discussed during this research effort is the use of supervisory control systems with treemap interfaces. The treemap could in theory represent the health of various indicators. For instance, element size may represent production inventory with the green/red scale depicting performance versus plan. The manager could then click on critical areas to drill down to further information. This approach is quite similar to the “Map of the Market” treemap developed by Smartmoney.com [37]. Finally, Asahi et al. discuss interactive versions of treemaps, where users could “pump up” the relative weighting of particular attributes, in effect increasing the size of elements relative to their peers [1]. Such an approach is not unlike the slider scalars used in the WVT.

REFERENCES

1. Asahi, Toshiyuki, David Turo, and Ben Shneiderman. "Using Treemaps to Visualize the Analytic Hierarchy Process." Human-Computer Interaction Laboratory, University of Maryland paper CS-TR-3293 (1994).
<http://www.cs.umd.edu/hcil/treemap3>
2. Berglund, C. Neil. "A Unified Yield Model Incorporating Both Defect and Parametric Effects." *IEEE Transactions on Semiconductor Manufacturing*, vol. 9, no. 3 (August 1996), 448.
3. Bertin, Jacques. *Graphics and Graphic Information Processing*. New York: de Gruyter, 1981.
4. Bruls, M., K. Huizing, and J.J. van Wijk. "Squarified Treemaps." *Proceedings of the joint Eurographics and IEEE TCVG Symposium on Visualization* (2000), 33-42.
5. Campbell, Stephen A. *The Science and Engineering of Microelectronic Fabrication*. New York: Oxford University Press, 1996.
6. Card, S., J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. San Francisco: Morgan Kaufmann Publishers, 1999.
7. Eberts, Ray E. *User Interface Design*. Englewood Cliffs, NJ: Prentice Hall, 1994.
8. Einhorn, Bruce. "Betting Big on Chips." *Business Week* (April 2001), 54-56.
9. Fine, Charles. *Clockspeed*. Reading, MA: Perseus Books, 1998.
10. Harvard Business School. "Intel-PED." Boston, MA: HBS Case Services (9-693-056).
11. Hayes, Robert, Steven Wheelwright, and Kim Clark. *Dynamic Manufacturing: Creating the Learning Organization*. New York: The Free Press, 1988.
12. Hopp, Wallace J. and Mark L. Spearman. *Factory Physics*. New York: McGraw-Hill, 2001.
13. Kline, Kevin, Lee Gould, and Andrew Zanevsky. *Transact-SQL Programming*. O'Reilly and Associates, 1999.
14. Kofler, Michael. *Definitive Guide to Excel VBA*. New York: Springer-Verlag, 2000.
15. Leachman, Robert C., John Plummer, and Nancy Sato-Misawa. "Understanding Fab Economics". *Competitive Semiconductor Manufacturing Program*, University of California at Berkley, Report CSM-47 (1999).

16. Mackinlay, J. "Opportunities for Information Visualization." *IEEE Computer Graphics and Applications* (Jan/Feb 2000), 22-23.
17. Nahmias, Steven. *Production and Operations Analysis*. 3rd edition. The McGraw-Hill Companies Inc., 1997.
18. Norman, Donald A. *The Psychology of Everyday Things*. New York: Basic Books, 1988.
19. Page, Alison L. "Forecasting Mix-Sensitive Semiconductor Fabrication Tool Set Requirements Under Demand Uncertainty." Cambridge, MA: MIT Thesis, 2001.
20. Powersim, "Powersim Methodology." White paper (2001).
<http://www.powersim.com>
21. Robertson, G. C., J. D. Mackinlay, and S. K. Card. "Cone Trees: Animated 3D Visualizations of Hierarchical Information." *Human Factors in Computing Systems Conference Proceedings (CHI'91)*, 189-194.
22. Rosenfield, Don. "Demand Forecasting." *The Logistics Handbook* (Chap 14), editors Robeson and Copacino. New York: The Free Press, 1994.
23. Schwartz, Peter. *The Art of the Long View: Planning for the Future in an Uncertain World*. New York: Currency Doubleday, 1996.
24. Scott Morton, Michael S. *Management Decision Systems*. Harvard University Press, 1971.
25. SCP Team. "SCP Objectives." Presentation Notes, November 2001.
26. Shneiderman, B. and Martin Wattenberg, "Ordered Treemap Layouts." *Human-Computer Interaction Laboratory, University of Maryland* (2001).
<http://www.cs.umd.edu/hcil/treemaps>
27. Shneiderman, B. "Treemaps for Space-Constrained Visualization of Hierarchies." *Human-Computer Interaction Laboratory, University of Maryland* (2001).
<http://www.cs.umd.edu/hcil/treemaps>
28. Spence, Robert. *Information Visualization*. New York: ACM Press, 2001.
29. Sprague, Ralph H. and Hugh J. Watson. *Decision Support Systems*. 2nd Edition. Englewood Cliffs, NJ: Prentice Hall, 1989.
30. Sterman, John D. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. Boston, MA: Irwin McGraw-Hill, 2000.

31. Tufte, Edward R. *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press, 1983.
32. University of Maryland, Human-Computer Interaction Lab. "Treemap 3.0 Documentation: Introduction to Treemap." (2002).
<http://www.cs.umd.edu/hcil/treemap3/tutorialdoc.shtml>
33. Van Zant, Peter. *Microchip Fabrication*. New York: McGraw-Hill, 1997.
34. Ventana Systems Inc., "Seeing Simulation with Synthesim." (2002).
<http://www.vensim.com>
35. Vining, G. G. *Statistical Methods for Engineers*. Pacific Grove: Duxbury Press, 1998.
36. Ware, Colin. *Information Visualization: Perception to Design*. Boston, MA: Morgan Kaufman Publishers, 2000.
37. Wattenberg, M. "Map of the Market." *SmartMoney.com* (1998).
<http://smartmoney.com/marketmap>